高等院校信息技术规划教材

PHP Web和Android开发入门与实践

邢益良 裴云 陈敏 编著

马亮 主审

清华大学出版社

高等院校信息技术规划教材

PHP Web和Android开发
入门与实践

邢益良　裴　云　陈　敏　编著

马　亮　主审

清华大学出版社

北　京

内 容 简 介

本书详细介绍 PHP 及其在 Web 和移动应用的基础知识。全书共 19 章，第 1～12 章介绍 PHP Web 开发，第 13～19 章介绍 PHP 移动开发。在 PHP Web 开发方面，首先系统地阐述 PHP 的基础知识，包括开发环境、变量、运算符、控制结构语句、函数、数组，以及 PHP 5 最新融入的面向对象技术中的类和对象，然后重点讲解 MySQL 数据库编程、表单获取、会话管理、XML、JSON、Xajax 和多媒体等 PHP Web 开发常用技术，最后推出投票系统和 Web 聊天室系统两个经典的 PHP 项目，对关键技术、设计方案、代码实现和运行结果等都作了较详细的分析。在 PHP 移动开发方面，首先介绍普通手机上的 WAP 基础知识和 PHP WAP 开发技术，然后较全面地阐述 Android 的基础知识，包括发展史、体系结构、开发环境搭建和辅助工具等，最后介绍 PHP Android 开发技术，包括界面和事件处理、数据管理、电池管理、系统管理、电话通信、短消息通信、WiFi 通信、语音合成和多媒体开发等技术。

本书涵盖 PHP 常用知识点，每个知识点都有对应的翔实可运行的代码，附有详细注释、说明及运行效果图。

本书适合作为高等院校计算机、软件工程和软件技术专业大专生、本科生和研究生教材，同时可供对 PHP 比较熟悉的开发人员和研究人员等参考。

图书在版编目(CIP)数据

PHP Web 和 Android 开发入门与实践/邢益良，裴云，陈敏编著. —北京：清华大学出版社，2014
高等院校信息技术规划教材
ISBN 978-7-302-36458-0

Ⅰ. ①P… Ⅱ. ①邢… ②裴… ③陈… Ⅲ. ①网页制作工具－PHP 语言－程序设计－高等学校－教材 ②移动终端－应用程序－程序设计－高等学校－教材 Ⅳ. ①TP393.092 ②TP312 ③TN929.53

中国版本图书馆 CIP 数据核字(2014)第 095433 号

责任编辑：张 玥 战晓雷
封面设计：傅瑞学
责任校对：白 蕾
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**24.25　　**字　　数：**559 千字
版　　次：2014 年 8 月第 1 版　　**印　　次：**2014 年 8 月第1 次印刷
印　　数：1～　000
定　　价：　.00 元

产品编号：059151-01

序言

foreword

基于开放源代码的 PHP 从 1994 年诞生至今已被上千万个网站采用，全球知名的互联网公司 Yahoo!、Google、新浪、百度、腾讯、TOM 和 YouTube 等的网站均是 PHP 技术的经典应用。随着技术的发展，PHP 已经发展成为一个适合企业级部署的技术平台，IBM、Cisco、西门子和 Adobe 等知名公司均在普遍采用 PHP 技术。Android 是智能手机快速发展的重要引擎，其市场占有率超过70%。Android 开发主要面向 Java 开发语言，然而 SL4A 和 PHPFORANDROID 的出现，让 PHP 敲开了 Android 开发之门，这使得 Android 开发更简单和快速。随着技术的不断发展，相信 PHP 一定能更胜任智能手机应用开发，势必成为 Android 应用主流开发语言，打破 Java 一统 Android 手机天下的局面，成为推动智能手机高速发展的重要力量。

目前，PHP 培训机构和大学的 PHP 专业课程体系与教材等存在诸多问题，这使得 PHP＋Web＋Android 的学习变得困难。PHP 是当今的主流开发语言，市面上有关 PHP 和 Android 的教材，或者阐述的是使用 PHP 开发 Web 应用，或者阐述的是使用 Java 开发 Android 应用，而不是使用 PHP 开发 Android 应用。本书的编写目标就是让学习 PHP 更加简单和快速，让 PHP 学习者能快速地进入 Web 和 Android 开发领域，能较好地从事 PHP 相关工作。

本书基于 Web 和 Android 开发等工作岗位的要求，面向 Web 和 Android 开发入门者，通过通俗易懂的语言和丰富多彩的实例，详细介绍使用 PHP 进行 Web 和 Android 开发应该掌握的常用技术。本书采用先易后难的原则，先讲述 PHP 基础，再讲述使用 PHP 开发 Web 和 Android 应用常用技术。突出“基础、系统、实践和内容新颖”的特点，具体特色如下：

(1) 内容基础和系统。系统讲述 PHP 语言、Web 开发和 Android 开发等工作原理和开发技术。

(2) 内容新颖。应用 PHP 开发 Android 软件，打破 Java 一统 Android 手机天下的局面。

(3) 实用性和实践性强。以 phpDesiger 等为编程工具，通过实例分析和实现，使读者深刻理解 PHP Web 和 Android 开发工作原理，读者使用本书中的 PHP 技术能快速地完成项目实践。

(4) 附有代码。每个实例都附有源代码，可直接运行，以帮助读者学习。

本书能让读者以较短的学习周期和较低的学习成本轻松领会 Web 和 Android 程序开发的精髓，快速掌握移动电子商务和移动电子政务等应用的开发技能，使从事 PHP 产品开发的企业低成本、快速地融进 Android 技术升级产品，走向成功。

本书第 5、8、10、15、16、17、18 和 19 章由邢益良编写，第 2、3、4、6、7 和 9 章由裴云编写，第 1、11、12、13 和 14 章由陈敏编写。在本书编写过程中，作者参考了大量书籍、资料和网站，同时，也融入了作者在软件开发教学和科研中的经验。限于作者的学识水平，书中谬误之处在所难免，敬请读者不吝指正。

本书适用于 PHP Web 和 Android 初学者、大中专院校师生和工程技术人员等学习和参考。如果读者曾接触过 HTML 和 JavaScript，学习效果会更好。

作者

2014 年 6 月

目录 Contents

第 1 章 chapter 1

PHP 5 概述

1.1 什么是 PHP

PHP 全称为 PHP Hypertext Preprocessor，是一种用来创建动态交互性站点的强有力的服务器端脚本语言，是一种被广泛使用的开放源代码多用途脚本语言，尤其适用于 Web 开发，并可以嵌入到 HTML 中去。PHP 语言是最受欢迎的 Web 开发语言之一，它以学习简单、开发快速和性能稳定而备受 Web 开发人员的青睐，同时，它在移动应用开发领域还具有广阔的发展前景。PHP 不仅使用人员众多，数以万计的 Web 站点都用它构建，而且有强大的社区支持，这使得无论是用 PHP 开发 Web 应用，还是学习 PHP 语言，都会快速有效、事半功倍。在各种 Web 开发语言、框架和概念纷至沓来的今天，PHP 仍以其独特魅力吸引更多的 Web 开发人员学习和使用。PHP 的主要目标是让 Web 开发人员可以很快写出动态生成的网页，但 PHP 的功能远不止如此，开发人员应用 PHP 还可以开发桌面、WML 和 Android 等应用程序。下面先来看一个嵌入了 PHP 代码的 HTML 文档，通过它初次体验使用 PHP 开发 Web 程序。

【例 1-1-1】 （代码位置：\1\helloworld. php）

```
<html>
  <head><title>Welcome</title><head>
    <body>
      <?php
        echo"hello,world";
      ?>
    </body>
</html>
```

把程序复制到 PHP 环境中并使用浏览器打开该程序，该程序运行的结果是在网页上输出了一个内容为“hello，world”的字符串。

PHP 是一种语法简单、功能强大的网络编程语言，在语法格式上，借鉴了广泛流行的 C、Java 和 Perl 等编程语言的特点。尽管 PHP 非常类似于 C 语言，但比 C 语言更简单、易学和易用，因此特别适合于学习过 C 语言，有志于网络编程的人学习和使用。本书中的 PHP 程序就是指使用这种 PHP 语言编写的程序。它只能被服务器所解释执行，只能

运行于服务器端,用户在浏览器中看到的只是经服务器解释后返回给浏览器的HTML等客户端代码(可从浏览器的"查看→源文件"菜单中看到),这些代码由浏览器解释执行并把执行结果显示在浏览器窗口中形成网页。在客户端,用户看不到PHP代码,看到的是HTML等客户端代码,这也是PHP语言写成的代码比较安全的一个原因。

1.1.1 PHP的历史

PHP已经历过PHP 1.0、PHP/FI、PHP 3、PHP 4和PHP 5等多个版本。由于PHP 4及更低版本缺少面向对象支持等原因,而较新的PHP 6还没有正式发布,因此开发时基本不用这些版本。目前广泛使用的是PHP 5。下面是PHP各个版本的简单介绍。

1. PHP 1.0

PHP最初是1994年由丹麦程序员Rasmus Lerdorf创建的,刚开始只是一个简单的用Perl语言编写的程序,用来统计他自己网站的访问,后来又用C语言重新编写,包括可以访问数据库,1995年以Personal Home Page Tools (PHP Tools)开始对外发表第一个版本,Lerdorf写了一些介绍此程序的文档,并且发布了PHP 1.0。

2. PHP/FI

PHP 1.0提供了访客留言本、访客计数器等简单的功能。以后越来越多的网站使用了PHP,并且强烈要求增加一些特性,比如循环语句和数组变量等,在新的成员加入开发行列之后,PHP 2.0于1995年发布,第二版定名为PHP/FI(Form Interpreter)。PHP/FI加入了对mSQL的支持,从此确立了PHP在动态网页开发上的地位。到了1996年底,有15 000个网站使用PHP/FI;到了1997年,使用PHP/FI的网站数字已超过五万个。

3. PHP 3

1997年,Tel Aviv公司的Zeev Suraski和Andi Gutmans开始合作,自愿重新编写了底层的解析引擎,宣布正式接任构建和升级PHP的工作,并按照GNU标准重新规划名字的缩写为PHP Hypertext Preprocessor,其他很多人也自愿加入了PHP的相关工作,从此PHP成为了真正意义上的开源项目。PHP 3公开测试9个月后,于1998年6月,PHP.net网站发布了PHP 3声明。PHP 3提供大量实用功能,具有较好的可扩展性,具备坚实的底层架构,提供面向对象语法,并支持常用数据处理。PHP 3发布以后,用户数量才真正开始了飞涨。

4. PHP 4

Andi Gutmans和Zeev Suraski重写了PHP核心,新的PHP核心被称为Zend(以Zeev和Andi的名字命名)引擎,1999年,Zend公司发布PHP引擎,即ZendEngine,并于2000年5月正式发布新版PHP 4.0。有了Zend引擎的PHP 4具有强大的功能和优异

的性能。PHP 4 的性能较 PHP 3 有着显著的提高，相同的脚本在 PHP 4 中运行，最高可以有近 10 倍的性能提升。并且 Zend 提供了脚本优化器，其可以把源程序转为二进制编译代码，在提高性能的同时，也保护了程序源码不被暴露。在 PHP 4 中增加了对各种 Web 服务器（如 Apache、IIS/PWS 及 OmniHTTPd 等）的支持。此外，PHP 4 还增加了一些新的语言特性，如丰富的数组操作函数、完整的会话机制、对输出缓存的支持等。PHP 4 也对一些跨平台的技术提供了扩展支持，如对 Adobe PDF、SWF、Java、Microsoft.NET 等技术的支持。

5. PHP 5

尽管 PHP 4 的发展是如此迅猛，但较之于其他流行的开发语言还是缺乏一些关键的功能特性，比如，PHP 4 中的面向对象功能并不完善，也无法实现异常（Exception）的捕捉与处理，因此在一些特殊问题的处理上还是捉襟见肘的。2004 年 7 月，PHP 5 正式版本的发布标志着一个全新的 PHP 时代的到来。它的核心是第二代 Zend 引擎，并引入了对全新的 PECL 模块的支持。PHP 5 的最大特点是引入了面向对象的全部机制，并且保留了向下的兼容性。程序员不必再编写缺乏功能性的类，并且能够以多种方法实现类的保护。另外，在对象的集成等方面也不再存在问题。使用 PHP 5 引进了类型提示和异常处理机制，能更有效地处理和避免错误的发生。

6. PHP 6

目前，PHP 作为风靡全球的开源脚本语言，在数以万计的网站上运行着。尽管 PHP 6 至今尚未发布，但它将包含很多新的令人激动的特性以及一些语法方面的改进，这将会让面向对象开发更加简单。

正是因为 PHP 具有的上述诸多优点，使得国内最知名的网站中有多家使用了 PHP，如百度、腾讯、新浪、搜狐、网易、淘宝、雅虎中国、Tom 在线和易趣等。

1.1.2 PHP 的特点

PHP 之所以拥有众多的开发者是因为其具有非常鲜明的特点，其特点如下所述。

(1) 快速。PHP 5 是一种功能强大的 CGI 脚本语言，语法混合了 C、Java、Perl 和 PHP 式的新语法，执行网页的速度比 CGI、Perl 和 ASP 更快，这是它的第一个突出特点。

(2) 具有很好的开放性和扩展性。PHP 属于自由软件，源代码完全公开，任何程序员要为 PHP 扩展附加功能非常容易。

(3) 数据库支持。PHP 支持多种主流与非主流的数据库，如 mSQL、MySQL、Microsoft SQL Server、Solid、Sybase、ODBC、Oracle 和 PostgreSQL 等。其中，PHP 与 MySQL 是绝佳的组合，这种组合可以跨平台运行。由于 PHP 支持 ODBC（开放数据库连接标准），因此 PHP 可以连接任何其他支持该标准的数据库。

(4) 面向对象编程。PHP 提供了类和对象。为了实现面向对象编程，PHP 5 提供了新的功能和特性，包括对象重载、引用技术等。PHP 5 可以选择面向对象的方式编程，也可以选择面向过程的方式编程，或者两者混合的方式编程。

(5) 版本更新速度快。PHP 的更新速度要快得多,能达到每几周就更新一次。

(6) 具有丰富的功能。PHP 包括对象式的设计、结构化的特性、图形处理、XML 解析、数据库的处理、网络接口应用和安全编码机制等,PHP 几乎涵盖了所有网站的一切功能。

(7) 可伸缩性。传统方式上,网页的交互作用是通过 CGI 来实现的。由于 CGI 为每一个正在运行的 CGI 程序开一个独立进程,因此,CGI 程序的伸缩性不很理想。解决方法就是将经常用来编写 CGI 程序的语言的解释器编译进 Web 服务器(比如 JSP 等),PHP 就可以以这种方式进行安装,尽管很少有人愿意这样以 CGI 的方式安装它。相比之下,内嵌的 PHP 可以具有更高的可伸缩性。

(8) 跨平台。PHP 跨平台性非常好,例如在 Linux 平台和 Windows 平台上都可以运行。

(9) 支持多种网络协议。PHP 支持 LDAP、SNMP、POP3 和 HTTP 等大量协议。

1.1.3 Web 2.0 主流技术

Web 2.0 是 2003 年之后互联网的热门概念之一,但它没有一个很明确的定义。由于 Web 2.0 的页面采用了无刷新的 Ajax 技术,因此客户体验更流畅。典型的 Web 2.0 应用包括博客、RSS、百科全书(Wiki)、社会网络、P2P 和即时信息等。目前,用于 Web 2.0 开发的主流服务器端开发语言有 Java、C# 和 PHP,其中 C# 是微软公司提供的开发语言,它具有开发快速和维护简单等优点,但缺点是成本高和不开源等。下面对 Java 和 PHP 在 Web 2.0 中的应用做一个详细的比较分析,供大家了解其各自优劣势。

(1) PHP 是解释执行的服务器脚本语言,因为语法和 C 语言比较像,所以学过 C 语言的程序员可以很快地熟悉 PHP 的开发,有简单容易上手的特点。要掌握好 Java,由于需要先学好 Java 的语法和熟悉一些核心的类库,懂得面向对象的程序设计方法,所以 Java 不如 PHP 易学。

(2) Java 通过 JDBC 来访问数据库,通过不同的数据库厂商提供的数据库驱动方便地访问数据库。PHP 提供 MySQL 扩展库访问 MySQL 数据库,开发者也可通过 ADODB 等数据库抽象层类库访问多种数据库。例如,Web 应用从 MySQL 数据库转到 Oracle 数据库,Java 只需要做很少的修改,而 PHP 如果选用 ADOdb 类库就只需做少量修改,如果选用 MySQL 扩展库则需要大量修改。

(3) Java 和 PHP 都有很好的跨平台的特性,其相关技术平台几乎都可以在不作任何修改的情况下运行在 Linux 或者 Windows 等不同的操作系统上。PHP 最经典的平台组合是 PHP+MySQL+Apache。其平台非常适合开发中小型的 Web 应用,开发的速度比较快,而且所有的软件都是开源免费的,可以减少投入。Java 的 Web 应用服务器有免费的 Tomcat 和 JBoss 等应用服务器,但免费的服务器其用户体验并不太理想。如果需要更好的商业化的服务,则可以选择 WebSphere 和 WebLogic 等服务器,但其费用较高。

(4) PHP 开发的程序源代码都是公开的,他人拿到 PHP 开发的程序后都可以进行修改。Java 开发的程序,最终用户拿到的只是一些编译好的 class 类,无法看到完整的源代码,安全性高。

(5) 尽管 PHP 5 开始支持面向对象技术，但同时支持面向过程技术，而 Java 只支持面向对象技术，和 Java 相关的框架技术和系统分析设计也都支持面向对象。

综上所述，PHP 适合于快速开发中小型应用系统，开发成本低，能够对变动的需求作出快速的反应。而 Java 适合于开发大型的应用系统，应用的前景比较广阔，系统易维护，可复用性较好。还有，同样功能的系统用 Java 开发的系统要比 PHP 开发的系统价格要高。

1.1.4 移动开发主流技术

目前在 Web 2.0 的世界里，PHP 已经是一个不可忽视的强大力量，身为开源的先锋，已经为各种各样的开源产品作出了表率作用，用现在流行的话说，PHP 展现了草根的力量，星星之火可以燎原。根据维基百科的统计，有 75％的网站是由 PHP 语言构建的，包括众多的一流网站，如 Facebook、维基百科和雅虎，都是用以前被很多专家看不起的 PHP 来开发的，开源的产品成为主流已经成为可能。近几年，随着智能手机的普及以及手机硬件性能的不断提高，移动开发已经进入了一个新的阶段，PC 和 Web 上的众多软件移植到移动系统上已经是大势所趋。

移动应用开发平台是目前软件和互联网领域的热门技术。目前，市面上的智能手机以 iOS 和 Android 智能手机为主，基于 iOS 和 Android 这两类手机平台的各种游戏和应用不断涌现，手机应用平台已经成为继 Windows 和 Web 平台后又一个主流的软件技术平台。iOS 是苹果公司的操作系统，Android 是谷歌公司发布的操作系统，iOS 是封闭的操作系统，它能够集成软硬件以提供较好的用户体验，而 Android 是开源的，能整合更多软硬件以提供更强的功能。下面就 iOS 和 Android 的特点进行比较。

(1) Android 操作系统的设备数量最多，最大的制造商包括三星、摩托罗拉、HTC、索爱和宏碁等公司，iOS 操作系统的设备数量则次之。

(2) 在应用程序数量方面，iOS 操作系统的应用程序数量居首，其数量超过 50 万款，远远超过 Android 应用程序的数量。

(3) 在是否支持第三方面应用方面，iOS 平台审核较为严格，开发者做应用上线需要苹果公司通过审核，而 Android Market 允许大部分 Android 应用进入，如果想安装未经审核的 Android 应用，只需用户同意即可，不必通过专门的机构审核，其自由度高于 iOS 应用。

(4) 在 API 控制方面，iOS 的 API 是被苹果公司严格控制的，第三方面应用通常无法拿到所有的 API，这意味着系统级别功能(打电话和拦截短信等)只能由苹果公司实现。而 Android 是开源的，其允许自由替换系统组件，也没有强制限制 API，这意味着开发者可实现打电话和拦截短信等较底层功能。

(5) 在多任务方面，两者都支持多任务功能。

(6) 在导航方面，两者都支持导航功能，但 Android 更具优势，因为谷歌地图服务很优秀，Android 和谷歌地图服务结合提供了更佳的用户体验。

(7) 在搜索引擎方面，Android 与谷歌搜索服务整合，iOS 默认用谷歌搜索服务整合，但用户可选择雅虎和 Bing。

(8) 在开发语言方面,iOS 指定使用 Objective-C 语言,也可以使用 Python 和 Ruby 等编程语言;Android 主要使用 Java 开发语言,但不限于 Java,可以使用 Perl 和 JavaScript 等多种脚本语言。

Android 应用的主要开发语言是 Java,这让很多 PHP 开发者觉得非常遗憾。但是自从 ASE(Android Scripting Environment,Android 脚本环境)和 PHPFORANDROID 诞生后,Android 应用已不局限于 Java 编程语言,ASE 允许用户直接在 Android 设备上运行脚本,ASE 已经支持 Python 和 Perl 等脚本语言,如在 ASE 上安装 PHPFORANDROID 则允许用户直接在 Android 设备上运行 PHP 脚本,这使 PHP 走进 Android 移动开发领域成为可能。

通常情况下,每个移动平台是需要采用特定的编程语言进行应用开发的,每个移动平台会都提供相应的集成开发环境让开发者采用特定的编程语言开发移动应用程序。尽管 ASE 和 PHPFORANDROID 允许开发者使用 PHP 开发 Android 应用,但长期以来市面上都没有支持 Android 移动开发的 PHP 集成开发环境。这也就是说,PHP 开发者在此之前无法用自己熟悉的语言来开发移动应用。Zend Studio 是新一代的 PHP IDE,高效的开发和维护 PHP 代码是它的核心。2013 年年底,Zend 公司发布 Zend Studio 10 预览版,Zend Studio 10 预览版的发布改变了这种状况,让超过 500 万的 PHP 编程者可在 iOS、Android 和 Windows Phone 这些移动平台上设计和打造应用程序。相信不久的将来,Zend Studio 将成为 PHP 开发移动应用的主流开发工具。

Android 的核心是 Linux,而 Linux 是 PHP 在服务器市场的最佳搭档,从安全和开源的角度来看都将是未来手机平台的巨大推动力量。除此,PHP 还可作为智能手机客户端的后台大数据处理重要技术,同时,PHP 在非智能手机应用方面还有重要作用。这些都将成为 PHP 推动移动开发的重要力量,PHP 未来的移动应用市场和由此产生的高收入岗位空间,以及给个人创业带来的机会,都是不可小觑的。

1.2 搭建 PHP 5 开发环境

1.2.1 PHPnow 集成开发套件

1. PHPnow 简介

PHPnow 是 Win32 下绿色免费的 Apache+PHP+MySQL 环境套件包。它安装简易,能快速搭建支持虚拟主机的 PHP 环境。附带 PnCp. cmd 控制面板,可帮助用户快速配置用户的套件,使用起来非常方便。PHPnow 是绿色的,解压后执行 Setup. cmd 初始化,即可得到一个 Apache + PHP + MySQL 环境。安装此环境后就可以直接安装 Discuz!、PHPWind、DeDe 和 WordPress 等重要程序。PHPnow 套件包内含有如下软件包:

(1) Apache-2.0.63 / 2.2.16;

(2) PHP-5.2.14;

(3) MySQL-5.0.90 / 5.1.50；

(4) Zend Optimizer-3.3.3；

(5) phpMyAdmin-3.3.7；

(6) eAccelerator 0.9.6-1。

phpMyAdmin 是一个用 PHP 编写的，可以通过 Web 方式控制 phpMyAdmin 和操作 MySQL 数据库的软件。通过 phpMyAdmin 可以完全对数据库进行操作，例如建立、复制和删除数据等。如果使用这样合适的工具，那么 MySQL 数据库的管理就会变得相当简单。

在安装 PHPnow 时，如果之前已经装有同类软件，例如操作系统中已安装 Apache 或 MySQL 软件，请先停止或卸载，否则会占端口。为避免端口冲突，最好关闭迅雷或修改迅雷的 BT 端口。PHPnow 的安装主要包括以下 3 个步骤。

(1) 解压：解压 PHPnow 套件到任意目录(不能含有中文！例如“桌面”)。

(2) 安装：运行套件中的安装程序 Setup.cmd，根据向导进行初始化。

(3) 使用：\htdocs 为网站主目录，执行 PnCp.cmd 进行设置或管理 PHP 环境。

注意：执行套件中的程序 Stop.cmd 即卸载 PHP 环境。请执行 Stop.cmd 后再删除前述解压文件。使用 PHPnow 完全免费。PHPnow 是一个集成套件，各组件的许可协议是独立分开的。为尊重 PHPnow，发布修改版和引用 PHPnow 等，应该要保留 PHPnow 官网。

2. 安装 PHPnow

搭建 PHP 开发环境其难度并不高，但比较烦琐。从头开始搭建 PHP+MySQL 环境是费时的，更糟的是，很多新手在配置 PHP 环境时常常出现这样那样的问题，诸如，MySQL 扩展和 zend 安装失败等问题。PHPnow 是一个快速、标准且专业的 PHP 套件包，它能够为用户快速搭建专业的 PHP+MySQL 开发环境。下面是 PHPnow 1.5.6 版本的搭建过程。

(1) 把从 http://www.phpnow.org/download.html 下载的 PHPnow-1.5.6.zip 解压到指定要安装的目录，例如下载解压到 c:\php 目录下，解压后目录结构如图 1-2-1 所示。



图 1-2-1 PHPnow 解压后目录

(2) 解压后执行 Setup.cmd,用户根据其提示进行安装,之后程序将会调用 Init.cmd 自动完成初始化。成功初始化后 Init.cmd 自动改名为 Init.cm_。如有必要,可将其改名为 Init.cmd 重新初始化。重新初始化不会丢失网站数据,仅仅是配置复位。执行 Setup.cmd 其结果如图 1-2-2 所示。



图 1-2-2 Setup.cmd 执行界面

(3) 根据提示输入 20 并回车,表示选择安装 Apache 2.0.63 Web 服务器,选择 22 表示安装 Apache 2.2.16 Web 服务器,安装过程如图 1-2-3 所示。

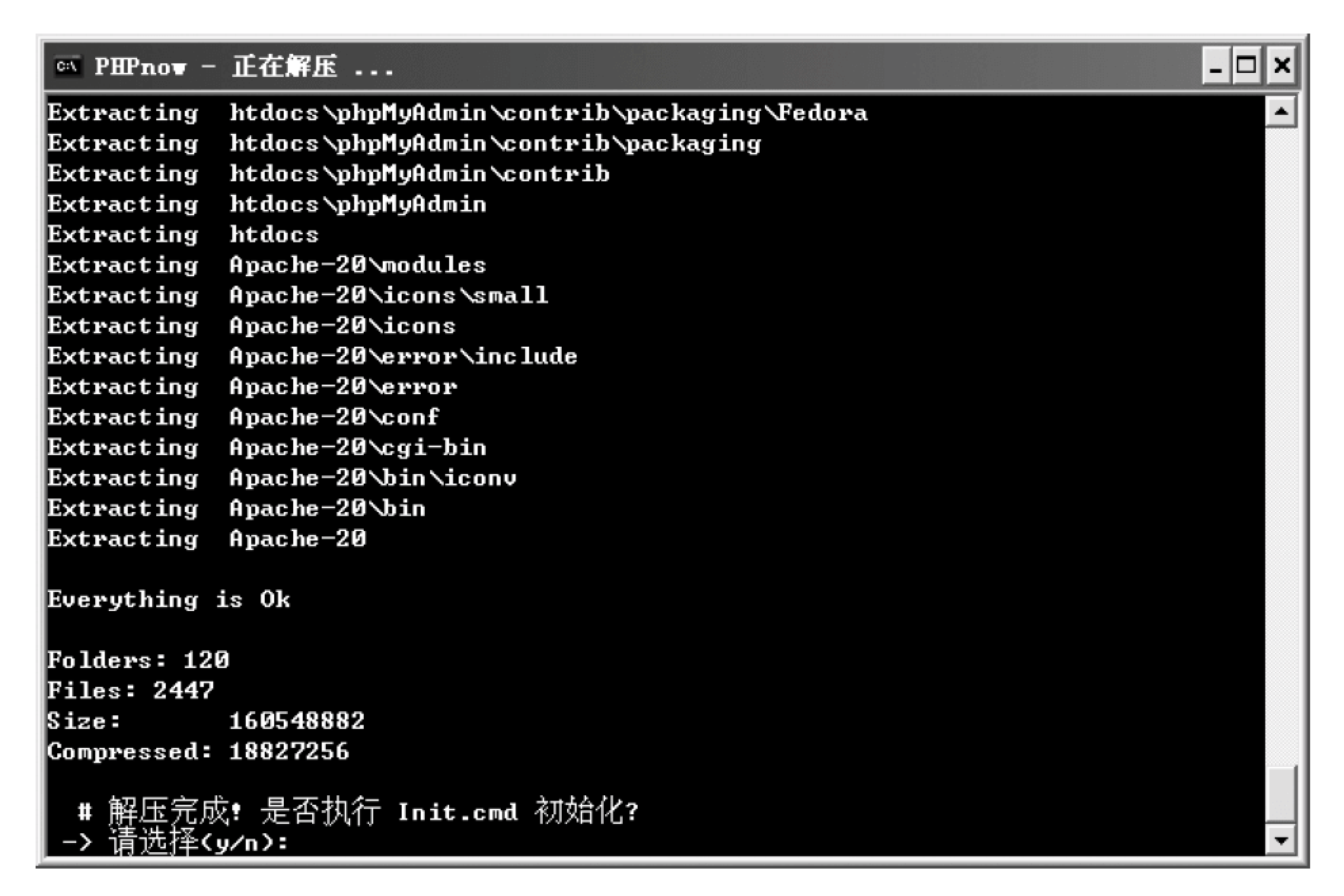


图 1-2-3 解压安装过程

(4) 安装 Apache Web 服务器后系统提示是否需要初始化,输入 Y 表示初始化。Apache Web 服务器成功初始化后,系统要求用户输入 MySQL 数据库的初始密码,结果如图 1-2-4 所示。

(5) 输入 MySQL 超级用户 root 的密码,例如输入的是 111111,执行完这一步后 PHPnow 安装文件目录结构将发生变化,其结果如图 1-2-5 所示。

到这里 PHP 的环境就搭建完成了,该环境内含有 Apache Web 服务器、PHP 引擎和 MySQL 数据库系统。为了证明环境是否已经搭建成功,可在浏览器中输入 http://127.0.0.1,浏览器将自动打开默认页,如图 1-2-6 上半部所示。在默认页 MySQL 用户密码文本框中输入前面设置的 MySQL 用户密码(111111),并单击"连接"按钮,表示使用 root 超级用户连接 MySQL 的默认数据库 test,如果系统安装成功则会在页面中显示 OK,运行结果如图 1-2-6 下半部分所示。

把 PHP 文件和 HTML 等文件复制到 c:\php\PHPnow-1.5.6\htdocs 文件夹,该文件夹表示 PHP Web 程序部署目录,凡是需要运行的程序都应复制到该文件夹,复制后即可通过浏览器访问。例如,可在 IE 网址栏输入 http://127.0.0.1/MyIndex.php,其表示访问网站根目录上的 MyIndex.php 页面。

```
初始化 PHPnow 1.5.6-1 - Apache 2.0 + PHP + MySQL 5.0

    开始文件处理 ...

已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。
已复制         1 个文件。

    文件处理完成;

    正在安装 Apache ...

    正在启动 Apache ...

    启动 Apache 完成;

    正在启动 MySQL 5.0 ...
Service successfully installed.
MySQL5_pn 服务正在启动 .
MySQL5_pn 服务已经启动成功。

    启动 MySQL 5.0 完成;

    现在为 MySQL 的 root 用户设置密码. 重要! 请切记!

 -> 设置 root 用户密码: 123456_
```

图 1-2-4　Apache Web 服务器安装和输入 MySQL 初始密码

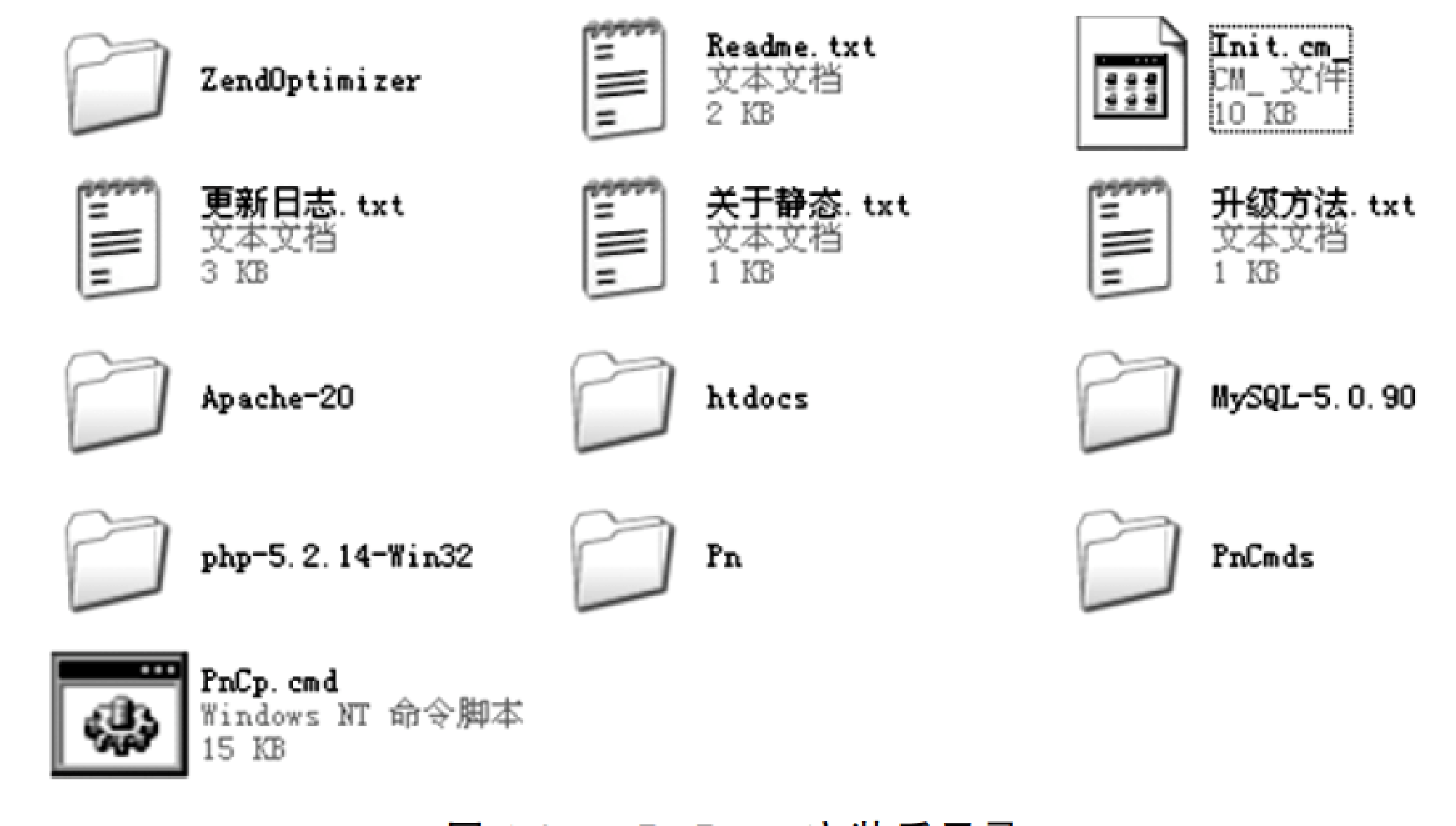


图 1-2-5　PHPnow 安装后目录

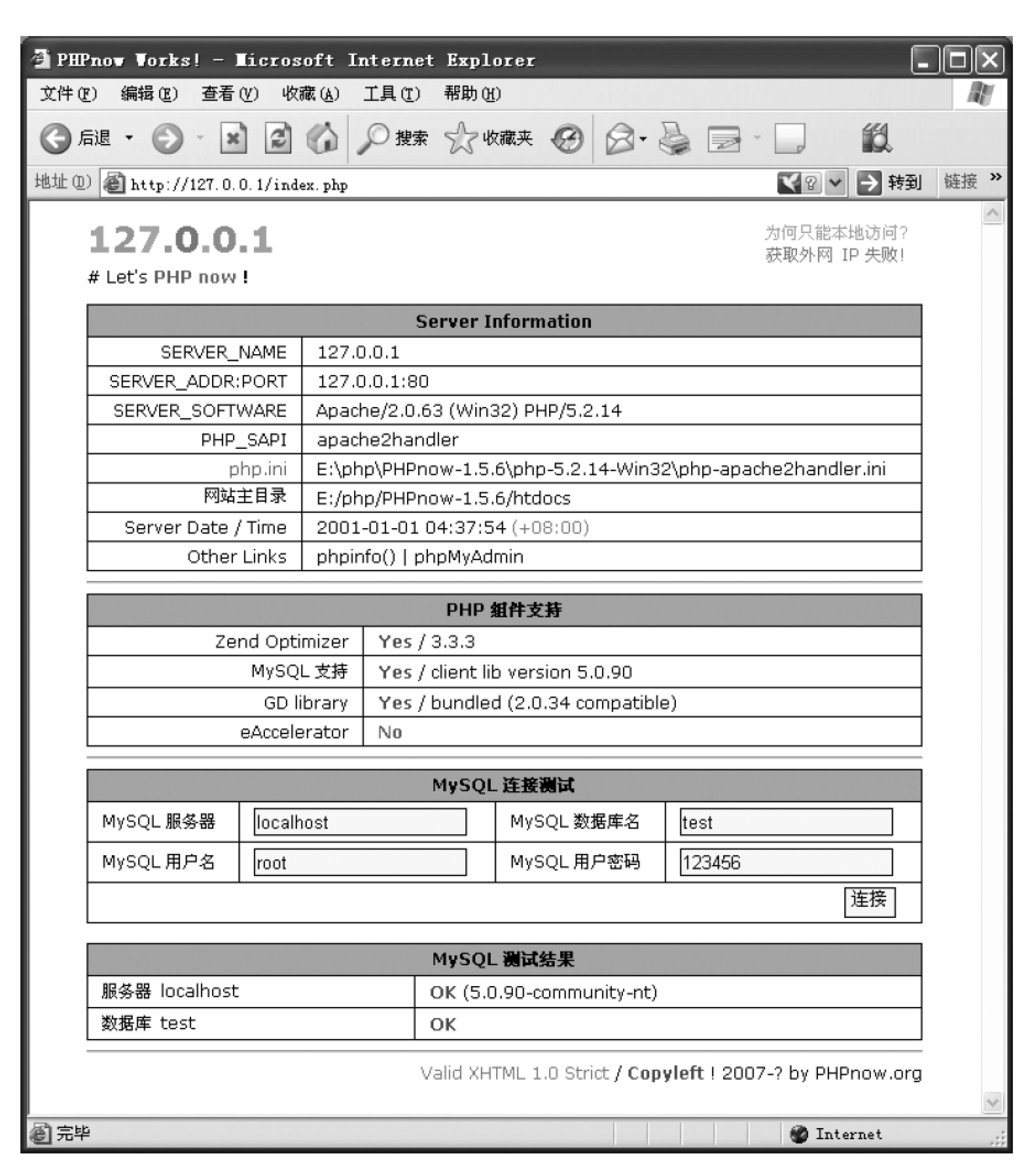


图 1-2-6　默认页面

如果要卸载 PHPnow，先执行目录 c:\php\PHPnow-1.5.6\PnCmds 中的 Stop.cmd 文件，然后再把之前解压的文件夹删掉即可。文件 Stop.cmd 在 PHPnow 安装目录下，如图 1-2-7 所示。



图 1-2-7　卸载 PHPnow

到此，一个专业的用于开发 Web 2.0 应用程序的 PHP 开发环境已经搭建成功。PHP 在 Android 中的开发环境搭建在后面章节中有详细介绍。

1.2.2 phpDesigner 工具

1. phpDesigner 简介

PHP 集成开发工具较多，主流的工具主要有 Zend Studio、EclipsePHP Studio、NetBean 和 phpDesigner 等。EclipsePHP Studio 简称 EPP，EPP 使用 Eclipse 核心编译，它是一个优秀的 PHP 集成开发软件，它集成了 JDK 和简体中文系统，免除了安装配置的麻烦，一次安装即可使用，无须配置，Eclispse 和 NetBean 更常用于 Java 开发。Zend Studio 是一个屡获大奖的专业 PHP 集成开发环境，非常适用于商务开发，但对 PHP 原生态初学者来说过于复杂。如果选用 Notepad＋＋工具，那么它的功能就显得不够，只适用于编辑 PHP 代码而不能运行和调试 PHP 程序。phpDesigner 是使用 PHP 开发 Web 应用较好的集成开发工具。phpDesigner 不仅是一个功能强大、性能超快的 PHP 集成开发工具，也是一个功能丰富的 CSS 和 JavaScript 编辑器，它既适合新手，也适合经验丰富的开发人员。phpDesigner 能帮用户编辑分析、测试和发布由 PHP、HTML 和其他 Web 语言驱动的 Web 应用程序，不管是 PHP 网页设计高手，还是刚入门的新手，都可以使用此软件来设计网页程序。较新版 phpDesigner 2008 是完整的集成开发环境，它不仅支持 PHP，也支持其他网络语言，包括 htm5l、XHTML、XML、CSS3 和 JavaScript 等开发语言。phpDesigner 2008 不仅支持 PHP 框架，也支持任何流行的 JavaScript 框架，如 jQuery，Ext JS、YUI、Dojo、MooTools 和原型。phpDesigner 支持 XDebug 用于调试 PHP 程序，同时，phpDesigner 还内建支持 SVN 版本管理功能。有关更多的 phpDesigner 内容，可到官方站点 http://www.mpsoftware.dk/phpdesigner.php 上了解。本书使用 phpDesigner 作为 PHP 开发工具。

2. phpDesigner 安装和使用

首先从 http://www.mpsoftware.dk/上下载 phpDesigner 工具，然后解压到硬盘，最后运行 phpDesigner.exe 即能打开 PhpDesigenr 软件，解压后目录结构和软件界面分别如图 1-2-8 和图 1-2-9 所示。

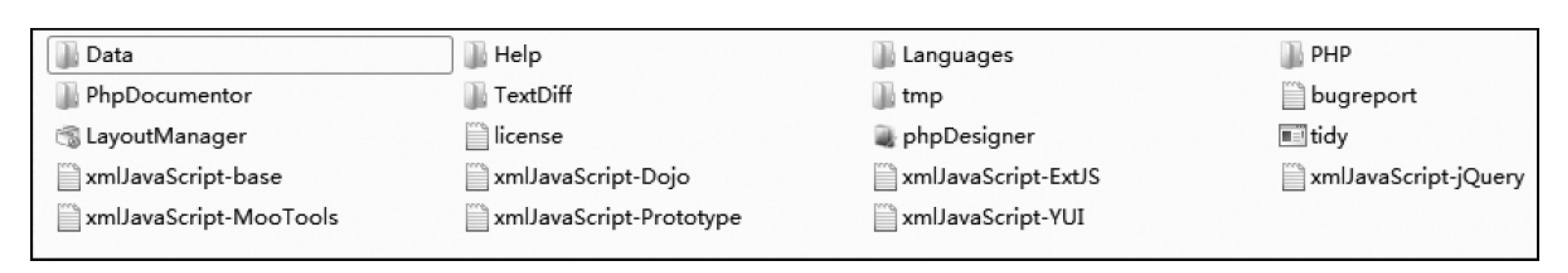


图 1-2-8 phpDesigenr 目录结构

phpDesigner 默认语言是英文，如果想用中文显示界面，可选择“查看”→“语言”→“中文”菜单命令，界面将用中文显示，操作过程如图 1-2-10 所示。

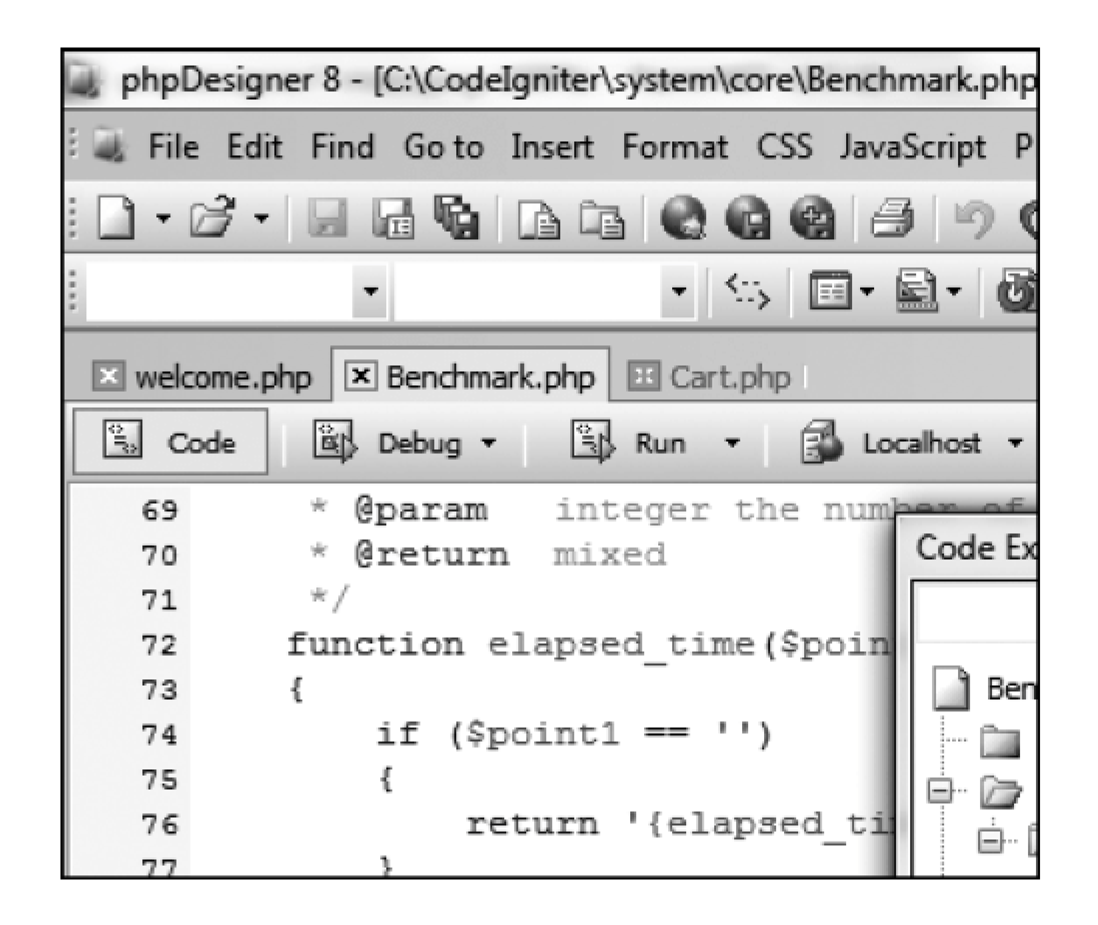


图 1-2-9 phpDesigner 界面



图 1-2-10 设置中文界面

1.2.3 新建 PHP 经典程序

当使用 phpDesigner 新添和编写 PHP 程序时，可选择“新建”→“新建”菜单命令，系统将弹出对话框，如 1-2-11 图所示，选择 PHP，再单击 Ok 按钮，系统将新建 PHP 文件，如图 1-2-12 所示，至此就可以开始编写 PHP 程序了。

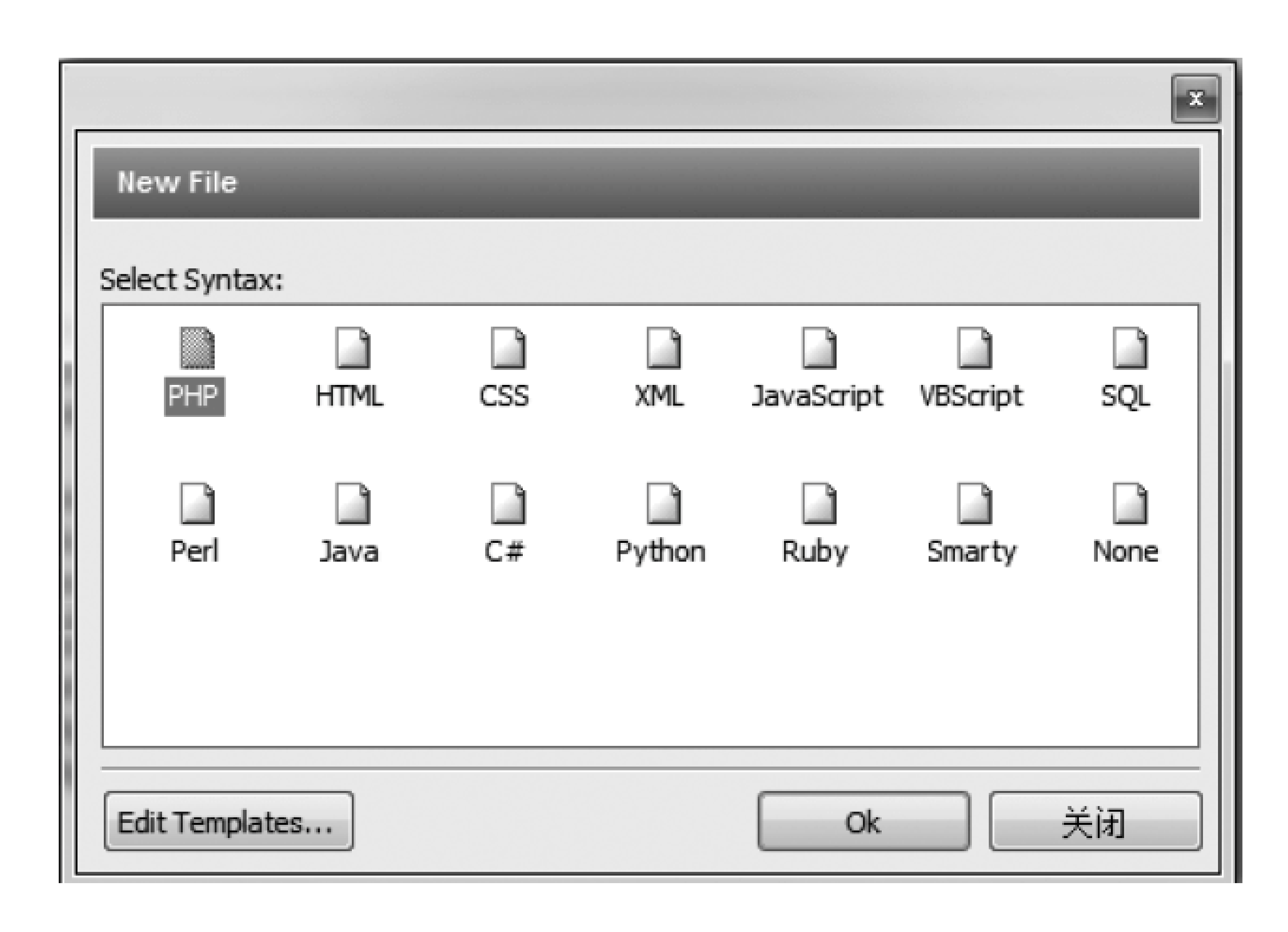


图 1-2-11 新建文件选项



图 1-2-12 phpDesigner 新建的 PHP 文件

在新建的 PHP 文件中输入如下代码。

【例 1-2-1】 (代码位置:\1\Hello.php)

```
<html>
  <head><title>Welcome</title><head>
    <body>
      <?php
        echo"Welcome to PHP's world!";
      ?>
    </body>
</html>
```

1.2.4 部署和运行 PHP 程序

如果想运行上述代码,则需把程序部署到 Apache Web 服务器中去。选择“文件”→“另存”菜单命令,给文件起名为 Hello.php,并存到 C:\php\PHPnow-1.5.6\htdocs(C:\php\PHPnow-1.5.6 目录是之前安装 PHPnow 的目录)。在 Apache Web 服务器运行的情况下(可通过控制面板中的管理工具中的服务功能查看名称为 Apache_pn 的服务),打开浏览器输入 http://127.0.0.1/Hello.php,浏览器将显示出“Welcome to PHP's world!”,结果如图 1-2-13 所示。

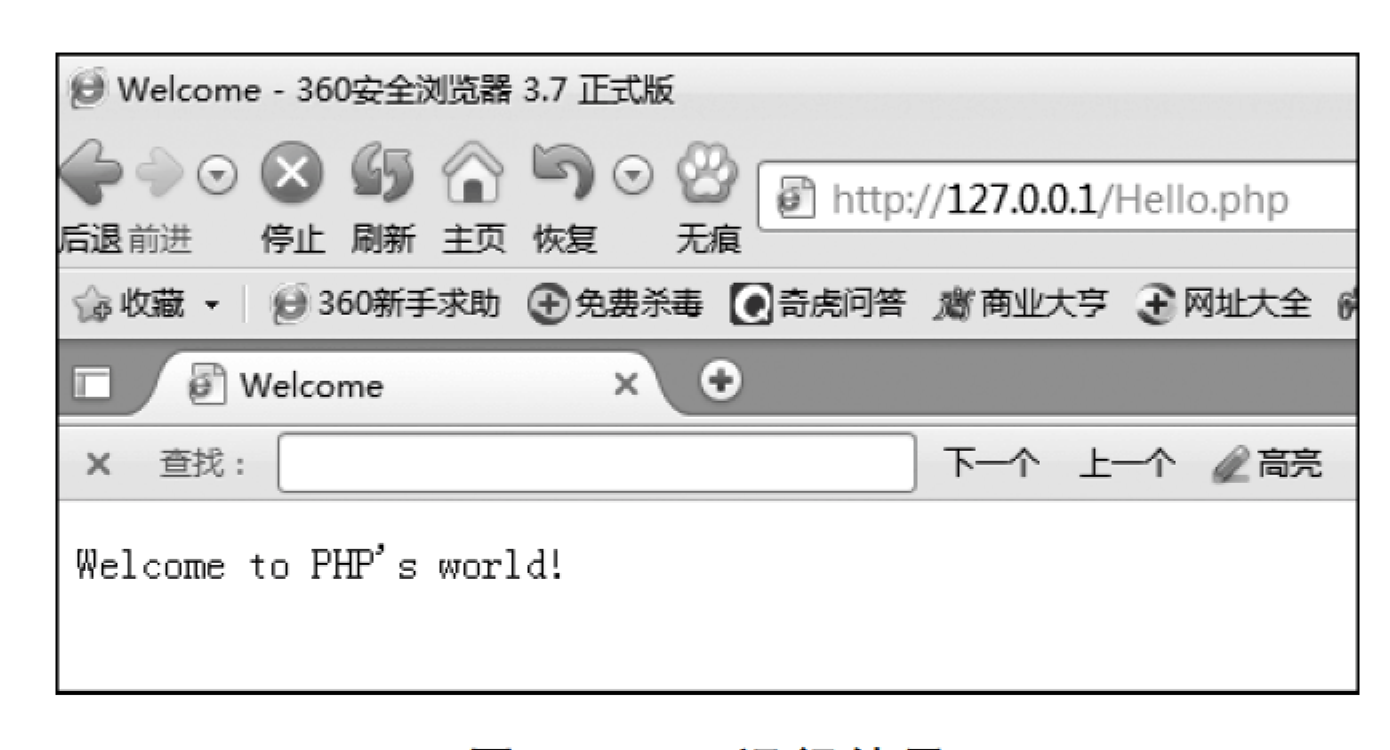


图 1-2-13 运行结果

第 2 章



PHP 5 开发基础

2.1　PHP 初探

2.1.1　典型的 PHP 程序

【例 2-1-1】（代码位置：\2\testhelloworld.php）

```
<?php
  //输出变量 a 的值
  $a="hello,world";
  echo $a;
?>
```

上例代码是典型的 PHP 程序，程序运行后字符串"hello，world"将显示在浏览器中，下面对这个 PHP 程序进行剖析。

（1）PHP 语言标记符。"<?"是 PHP 语言开始标记符，"? >"是结束标记符，二者之间的部分是 PHP 代码。需要注意的是：①PHP 语言标记符有多种，"<?"和"? >"是最为精简的一种，为简便起见，本书采用这种标记符；②"<?"和"? >"中的 PHP 代码与"<?"以及"? >"至少应有一个空格或回车，以便 PHP 语言解释器能正确区分它们。

（2）变量。$a 是 PHP 变量。变量名区分大小写。变量的表示形式为一个美元符号跟一个变量名称。例如，$a、$_a 和 $_a123_123_a 是合法变量，而 $123a 是不合法变量。变量使用前不需要声明数据类型，其类型由所存储的数据自动决定。

（3）字符串常量。"hello，world"是 PHP 字符串常量。

（4）赋值运算。符号"="是赋值运算符，其表示把"="符号右边的值存储到左边的变量中。

（5）语句结束符。分号";"是语句结束符，这一点和 C 语言相同。

（6）关键字。echo 是 PHP 的关键字，用于输出变量存储的内容。

（7）注释。"//"所在行是注释行，其采用的是 C/C++ 语言形式的注释方法，常用于简短的行注释。行注释也可采用 UNIX Shell 语言的注释方法，即把"//"换成"#"即可。多行注释的方法与 C 语言同，用"/* 注释内容 */"的形式进行表示。

这个程序虽然简单，但基本浓缩了 PHP 的语法要素，其形式与 C 语言类似甚至更简单。

上述程序先给变量 $a 赋值“hello,world”,然后通过 echo 语句把变量 $a 中的值显示出来。

2.1.2 PHP 代码嵌入到 HTML 文件

在 Web 应用开发中,PHP 程序通常会和 HTML 混合编写。当 PHP 与 HTML 语言混合开发时,可以使用 4 种不同的 PHP 语言标记符把 PHP 代码嵌入到 HTML 代码中。这 4 种标记符分别是:①<?php…?>;②<script language="php">…</script>;③<?…?>;④<%…%>。前两种方法总是可用的,而后两种方法要起效需要在配置文件(php-apache2handler.ini)中修改相应的指令,分别使配置文件中的指令 short_open_tag=on 和 asp_tags =on,最后一种是 ASP 的书写风格。在一个 PHP 应用中,多个不同的 PHP 文件应该只使用同一种标记保持相同的编写风格。下面是 4 种不同书写风格的 PHP 代码。

1. 第一种风格

```
<?php
  echo '若要服务于 XML 或 XHTML 应用,请采用这种标记符';
?>
```

2. 第二种风格

```
<script language="php">
  echo '有些编辑器如 FrontPage 不支持这种标记格式';
</script>
```

3. 第三种风格

```
<? echo '这是最简练的标记格式'; ?>
```

4. 第四种风格

```
<%echo '这是 ASP 风格的标记格式'; %>
```

在使用这 4 种风格编写 PHP 代码时,以下几点是初学者容易犯的错误,需要特别注意。

(1) PHP 代码与 PHP 语言开始标记符和结束标记符之间,至少应有一个空格或回车,以便 PHP 语言解释器能正确区分它们。

(2) 如果将 PHP 嵌入到 XML 或 XHTML 中则需要使用<?php…?>以保持符合标准。

(3) 开发需要发行的程序或者库,或者在用户不能控制的服务器上开发,应避免使用短标记,因为目标服务器可能不支持短标记,同时,为了代码的移植及发行,确保不要使用短标记。

(4) 凡嵌有 PHP 代码的程序文件，应具有 Apache Web 服务器能正确识别的 PHP 文件扩展名(在 Apache 配置文件中可查)，以便 Apache Web 服务器能将其提交给 PHP 去解释执行。

下例是将 PHP 代码嵌入到 HTML 代码中的范例，范例的任务是将表 2-1-1 中的学号和成绩以表格形式输出到浏览器，表格中的数据可由 PHP 代码生成，数据显示由 HTML 代码完成。

表 2-1-1　成绩表

序 号	学 号	成 绩	序 号	学 号	成 绩
0	1	1	3	4	4
1	2	2	4	5	5
2	3	3			

【例 2-1-2】 (代码位置：\2\tetsHTMLPHP.php)

```
<?
  //查询数据库,将查询结果保存到数组$cj
  //以上过程代码略
  //为演示方便,用下面直接赋值的方式代替以上过程
  $cj=array(0=>array("1","1"),
            1=>array("2","2"),
            2=>array("3","3"),
            3=>array("4","4"),
            4=>array("5","5"),
  );
?>
<html>
  <head>
    <title>hello,world</title>
  </head>
  <body>
    <center>查询结果</center>
    <table border=1 align=center>
      <tr><td>学号</td><td>成绩</td></tr>
        <? for($i=0;$i<count($cj);$i++){?>
          <tr><td><? echo $cj[$i][0];?></td><td><? echo $cj[$i][0];?></td></tr>
        <?}?>
    </table>
  </body>
</html>
```

将代码保存成文件 tetsHTMLPHP.php 并部署到 Web 服务器中，从浏览器向 Web 服务器发送查看该文件请求，其结果如图 2-1-1 所示。

查询结果

学号	成绩
1	1
2	2
3	3
4	4
5	5

图 2-1-1　成绩表

2.1.3 注释

注释是对程序文件或者代码片段额外添加的一个备注说明。通过注释可以提高代码可读性,让自己或其他开发人员更快速地理解程序内容。注释在程序编写中非常重要,恰当地注释程序不仅是程序员的个人习惯,更是责任的体现。但在实际过程中,大部分程序员都没有对程序进行注释的良好习惯,这往往会在日后给自己或他人造成相当大的麻烦。PHP 注释格式分为单行注释与多行注释。在 PHP 程序中,加入注释的方法很灵活。可以使用 C 语言、C++ 语言或者是 UNIX 的 Shell 语言的注释方式,也可以混合使用。这可以让每个写 PHP 网页程序的开发者发展出属于自己的写作风格。下面是 C++ 、C 和 UNIX Shell 语言风格的 PHP 注释的范例。

【例 2-1-3】 (代码位置:\2\testphpnote.php)

```
<?php
  echo "这是第 1 种方法。" ;        //这是 C++风格的单行注释
  echo "这是第 2 种方法。";
  /* 本例 C 风格的多行
          注释方式      */
  echo "这是第 3 种方法。";#本例使用 UNIX Shell 语法的注释
?>
```

2.2 变　　量

在程序中可以改变的数据量叫做变量,变量必须有一个名字,程序通过变量的名字来存储值,值可以是数字、文本字符串和数组等。一旦设置了某个变量,就可以在脚本中重复地使用它。程序可以通过赋值、打印输出和数学运算等操作符来使用变量。可以把变量看成是一个盒子,变量名看成是盒子的名称,变量值看成是盒子里存放的东西,这些东西的类型可以是肉、菜和书等。

2.2.1 变量的赋值

在 PHP 中,符号“=”不表示相等,而表示赋值。它的含义是将一个值指定给一个变量,更专业地说,它表示把“=”右边表达式的值赋给左边的变量,如“ $a=5”表示将 5 赋给变量 $a。赋值表达式的值也就是所赋的值,这就是说,“ $a=5”的值是 5。

PHP 赋值有两种方式:传值赋值和引用赋值。要掌握变量赋值通常需要先弄清传值赋值再弄清引用赋值。下例是传值赋值范例,程序先创建一个存有字符串的变量 $txt,通过“=”赋值号把字符串值“Hello World!”存到变量 $txt 中,然后再创建一个存有数值的变量 $number,把值 16 存到变量 $number 中,最后显示这两个变量的值。

【例 2-2-1】 (代码位置:\2\testVarValue1.php)

```
<?php
```

```
  $txt="Hello World!";
  $number=16;
  echo $txt;
  echo $number;
?>
```

下面是另一个传值赋值的范例,例中用变量 $a 而不是数字或字符串值给变量 $b 赋值。

【例 2-2-2】 (代码位置:\2\testVarValue2.php)

```
<?php
  //演示传值赋值
  $a=123;
  $b=$a;
  $a=321;
  echo "$a,$b";         //显示"321,123"
?>
```

引用赋值是指引用变量引用了源变量,改变引用变量(或源变量)的值将影响到源变量(或引用变量)的值,引用赋值需要将一个 & 符号加到源变量前。下面是引用赋值的范例。

【例 2-2-3】 (代码位置:\2\testVarReference.php)

```
<?php
  //演示引用赋值
  $a=123;
  $b=&$a;               //& 表示引用赋值,其含义是引用变量$b引用了源变量$a
  $a=321;               //对源变量$a的修改将影响到引用变量$b
  echo " $a,$b ";       //显示"321,321"
?>
```

传值和引用赋值的区别是:传值赋值中的变量互相独立,不会互相影响,而引用赋值中的变量互相影响,其值相同。

2.2.2 变量打印

有时候需要把变量的值打印输出到浏览器或另一个变量,或者供开发人员在调试时使用,或者为了给用户提供有用信息,PHP 提供了一些用来打印输出变量的语句或函数:echo、print、die、printf、sprintf、print_r 和 var_dump。

1. echo

echo 语句用于输出一个或多个字符串,其语法如下:

```
echo(strings)
```

参数 strings 是必需的,表示一个或多个要发送到输出的字符串。echo 实际上不是

函数而是语句,因此无须对其使用括号,同时,它执行后没有返回结果。如果希望向 echo 传递一个或多个参数,那么使用括号会发生解析错误。下面是使用 echo 打印多个变量和字符串的范例。

【例 2-2-4】 (代码位置:\2\testecho1.php)

```
<?php
  $a="How";
  $b=" do";
  $c=" you do";
  echo $a.$b.$c ;      //显示 3个变量的值,用点符号连接变量
  echo "<br>";
  echo "$a$b$c" ;      //显示 3个变量的值,用双引号
  echo "<br>";
  echo '$a$b$c';       //显示 3个变量的名称,用单引号
?>
```

程序运行结果如图 2-2-1 所示。

```
How do you do
How do you do
$a$b$c
```

图 2-2-1 运行结果

打印多个字符串变量的值可以使用点符号连接变量或者使用双引号包含变量。单引号和双引号的不同之处是:单引号仅输出变量名而不是值。但要注意的是,echo 可以使用括号,也可以不用括号。

```
<?php
  echo("hello");
?>
```

程序输出结果是 hello。

```
<?php
  echo"aaa",10,"bbb";
?>
```

程序输出结果是 aaa10bbb。

```
<?php
  echo("aaa",10,"bbb");
?>
```

程序运行时系统将输出错误提示。

2. print

函数 print 的作用是输出一个字符串,当执行失败会返回 false 值,其作用和 echo 相同,但执行速度比 echo 慢。

```
<?php
  print("hello php") ;
?>
```

程序运行结果：hello php。

3. die

函数 die 有两个功能：先输出内容，然后退出程序。下面是 die 函数的范例。

【例 2-2-5】 （代码位置：\2\testdie. php）

```
<?php
  $info='结束程序';
  die ($info);
  echo "你能看到这句话吗?";
?>
```

程序运行后只显示出字符串"结束程序"，因为 die 打印出字符串后就结束程序而不会往下继续执行。

die 常用在链接数据库中，比如：

```
mysql_connect("locahost","root","root") or die("链接服务器失败!")
```

如果连接数据库失败则显示字符串"链接服务器失败!"并结束程序的执行。

4. printf

函数 printf 用于输出格式化字符串，其语法如下：

```
int printf(string format, mixed [args]...)
```

参数 format 是转换的格式，以百分比符号%开始到转换字符为止。参数 args 是可选参数，表示需要打印输出的变量。函数返回值为整型。参数 format 的转换类型码如表 2-2-1 所示。

表 2-2-1　转换类型码

转换类型码	说　明	转换类型码	说　明
b	整数转成二进制数	o	整数转成八进制数
c	整数转成对应的	s	转成字符串
d	整数转成十进制数	x	整数转成小写十六进制数
f	单倍精度数字转成浮点数	X	整数转成大写十六进制数

下面是 printf 函数的范例，例中使用各种不同的转换类型码打印同一个变量，会打印出不同的结果。

【例 2-2-6】 （代码位置：\2\testprintf. php）

```
<?php
  $num=100.001;
  printf("%d",$num);          //100
  printf("%s",$num);          //100.001
  printf("%s---%d---%b---%x---%o---%f",$num,$num,$num,$num,$num,$num);
```

```
                                //100.001---100---1100100---64---144---1001.00100
  printf("%.2f",$num);          //100.00 (小数点保留 2 位)
  printf("%.1f",$num);          //100.0 (小数点保留 1 位)
  printf("%`#10s",$num);        //###100.001
  printf("%`@10s",$num);        //@@@100.001
  printf("%`_10s",$num);        //___100.001
  printf("%`#-10s",$num);       //100.001###
  printf("%`@-10s",$num);       //100.001@@@
  printf("%`_-10s",$num);       //100.001___
  printf("%`#20s",$num);        //#############100.001
?>
```

程序运行结果如图 2-2-2 所示。

```
100100.001100.001---100---1100100---64---144---100.001000100.00100.0#10s@10s_10s#-10s@-10s_-10s#20s
```

图 2-2-2 printf 运行结果

5. sprintf

sprintf 和 printf 很相似,不同的是,sprintf 并不把结果直接输出到浏览器,而是先赋给一个变量,然后再输出变量。

```
<?php
    $num=100.001;
    $a=sprintf("%d",$num);   //把结果输出到变量$a 中
    echo $a;                 //显示 100
?>
```

6. print_r

函数 print_r 的作用只用于输出数组,它带有一个数组参数。它常用于调试以便观察数组的键值。

```
<?php
  $arr=array(1,2,3,4);
  print_r($arr);
?>
```

程序输出结果为:Array([0]=>1[1]=>2[2]=>3[3]=>4)。

7. var_dump

var_dump 的功能是输出变量的内容、类型和长度,能让开发人员迅速了解一个变量,常用于调试。

```
<?php
  $a=100;
```

```
  var_dump($a);              //int(3) 100
?>
```

程序输出结果为 int(100)。

2.2.3 变量名命名

PHP 程序中的变量必须要有一个名字，程序通过变量的名字来存取值。一旦设置了某个变量，就可以在脚本中重复地使用它。PHP 中使用美元符($)后跟变量名来表示一个变量，如 $ var 就是一个变量。PHP 中的变量名是区分大小写的，$ var 和 $ Var 表示的是不同的两个变量。PHP 变量的命名需要遵守如下的规则：一个有效的变量名必须是 $ 符号为首，紧接 $ 符号的首个符号必须是英文字母或下划线，后面可以跟任意数量的英文字母、数字、下划线或其组合，变量名不能包含空格。$ abc、$ _ab_c、$ a1_b_c2 都是合法的变量名，而 $ 3xyz 和 abc 不是合法变量名。如果变量名由多个单词组成，那么应该使用下划线进行分隔(比如 $ my_skin)，或者单词以大写字母开头(比如 $ mySkin)。下面是合法和非法变量的范例。

【例 2-2-7】 (代码位置：\2\testValideVar. php)

```
<?php
  $var='Bob';                //合法变量名
  $Var='Joe';                //合法变量名
  echo "$var,$Var";          //输出两个变量的值
  $4site='not yet';          //出错!非法变量名,$符号后的首个字符是数字
  $_4site='not yet';         //合法变量名,$符号后的首个字符是下划线
?>
```

PHP 是一种松散类型的语言。在 PHP 中，不需要在设置变量之前声明该变量。下例是不声明数据类型范例，在该例中，可以看到不必向 PHP 声明该变量的数据类型。根据变量被设置的方式，PHP 会自动地把变量转换为正确的数据类型，有关 PHP 的数据类型在后面章节有详细介绍。在强类型的编程语言中(比如 C 语言)，必须在使用前声明变量的类型和名称。在 PHP 中，变量会在使用时被自动声明。

【例 2-2-8】 (代码位置：\2\testVarType. php)

```
<?php
  $var="Bob";                //无须声明变量的类型
  echo $var;                 //输出变量的值
?>
```

2.2.4 变量的作用域

变量的作用域指的是变量的有效范围，一个变量只能在其有效范围内才能存取它的值，如果超出有效范围，变量也就失去其意义了。PHP 提供 3 种不同作用域的变量：局部变量、全局变量和静态变量。

1. 局部变量

多数 PHP 变量都是局部变量，在其作用域外无法引用，并且超出其作用域时自动销毁。在当前文件主程序中定义的局部变量，其作用域限于当前文件的主程序，不能在其他文件或当前文件的局部函数中起作用；在局部函数中定义的变量，其作用仅限于当前函数，当前文件中主程序、其他函数和其他文件中无法引用。

2. 全局变量

作用域为程序级或文件级的变量称为全局变量。在函数中使用全局变量时，可以与当前文件的主体程序共同变量。全局变量必须使用 global 关键字表示，否则就视为局部变量。

3. 静态变量

静态变量仅在当前文件的当前函数中存在，但当程序执行离开此作用域时，其值并不丢失，当再次回到其作用域时，又可以继续使用原来的值；而一般变量是在调用结束后，其存储的数据值将被清除。静态变量须使用 static 关键字声明。

下面是局部变量的范例。

【例 2-2-9】 （代码位置：\2\testVarLocal. php）

```
<?PhP
  $a=10;
  function Test()
  {
    $a=20;                         //定义局部变量
  }
  Test();                          //调用函数
  echo $a;                         //变量$a未定义
?>
```

例中主程序和 Test 函数分别各自定义了变量 $a，主程序中的变量 $a 的值为 10，当主程序调用 Test 函数时，Test 中的局部变量 $a 被设置为 20，但它不会影响主程序中的变量 $a 的值，程序运行结果是 10。

下面是全局变量的范例。

【例 2-2-10】 （代码位置：\2\testVarGlobal. php）

```
<?php
  $zy="你好";
  $zyy="PHP语言";
  function lxt(){
    echo $zy;                      //$zy不能被调用，没有输出
    global $zyy;                   //利用关键字 global 在函数内部定义全局变量
    echo $zyy;
```

```
    }
    lxt();
?>
```

例中主程序中定义有变量$zy和$zyy。当函数lxt被调用时,由于$zy(作用域属主程序)在lxt中并未定义,所以会输出错误提示;而变量$zyy被关键字global声明为全局变量,不仅可以在主程序中使用,而且还可以在函数lxt中使用。程序运行结果如下:

```
Notice: Undefined variable: zy in C:\PHP\htdocs\testVarGlobal.php on line 5
PHP语言
```

下面是静态变量的范例。

【例2-2-11】 (代码位置:\2\testVarStatic.php)

```
<?PhP
function Test()
{
    static $a=10;                //定义静态变量
    echo $a;                     //输出变量
    $a++;                        //修改变量
}
Test();                          //调用函数
echo $a;                         //变量$a未定义
Test();                          //再次调用函数
?>
```

例中变量$a是静态变量,当函数Test被调用时,变量$a先被置为10,然后输出10,后变量自加操作,结束时变量$a的值是11;当函数Test再次被调用时,变量不会重新置为10,而是继续使用上次的值,即11,变量$a自加操作后值为12。程序运行结果如下:

```
10
Notice: Undefined variable: a in C:\PHP\htdocs\testVarStatic.php on line 9
11
```

2.2.5 表单变量

变量可以由用户自定义,也可以是PHP系统预定义的变量。变量$_GET、$_POST和$_REQUEST是系统预定义的,这些变量称为表单变量,因为它们的作用是获取网页表单中的信息,并把获取的信息传递给相应的页面进行处理,这是交互类网站设计的必用技术。可以将变量$_GET当做一个数组,用户在网页中填写的信息会通过HTTP的GET方法提交至PHP脚本,脚本会将用户填写的信息以数组的形式存储在变量$_GET中供程序使用。变量$_POST的作用和$_GET类似,不过,$_POST只接受由HTTP的POST方法提交的数据而不接受GET方法提交的数据。变量$_

REQUEST 包含 $ _GET 和 $ _POS 的内容，$ _REQUEST 变量可以用来获取通过 GET 和 POST 这两种方法发送的表单数据，或者说，变量 $ _GET 用来读取 GET 方法发送来的表单数据，变量 $ _POST 用来读取 POST 方法发送来的表单数据，而 $ _REQUEST 变量不需要区分这两种方法就能读取表单数据。

下面是关于变量 $ _REQUEST 的范例，该范例由文件 welcome. htm 和 welcome. php 组成，前者是界面，用于提交用户数据至程序 welcome. php，程序 welcome. php 会通过表单变量 $ _REQUEST 读取和处理用户数据。

【例 2-2-12】（代码位置：\2\welcome\welcome）

文件 welcome. htm 内容如下：

```
<form action="welcome.php" method="post">
姓名：<input type="text" name="name"/>
年龄：<input type="text" name="age"/>
<input type="submit" />
</form>
```

文件 welcome. php 内容如下：

```
你输入的姓名是：
<?php
echo $_REQUEST["name"];
?>
<br />
你输入的年龄是：
<?php
echo $_REQUEST["age"];
?>
```

用户在浏览器中打开 welcome. htm 页面后，在“姓名”和“年龄”文本框里输入“张三”和 30，如图 2-2-3 所示。



图 2-2-3　表单页面

单击“提交查询内容”按钮，welcome. php 页面通过 $ _REQUEST 读取“姓名”和“年龄”文本框中的内容并显示出来，运行结果如下：

```
你输入的姓名是：张三
你输入的年龄是：30
```

2.2.6　常量

函数 define 用于定义常量，它有两个参数，一个是常量名，一个是常量值。一个常量一旦被定义，就不能再改变或者取消定义。下面是定义常量的例子。

```
<?php
  define("CONSTANT", "你好!");        //CONSTANT是常量名,"你好"是常量值
  echo CONSTANT;
?>
```

常量名和其他任何PHP标签遵循同样的命名规则。合法的常量名以字母或下划线开始,后面跟着任何字母、数字或下划线。常量默认为大小写敏感,按照惯例常量标识符总是大写的。在脚本执行期间该值不能改变。定义常量和定义变量的区别如下:

(1) 常量前面没有美元符号($)。

(2) 常量只能用define函数定义,而不能通过赋值语句。

(3) 常量可以不用理会变量范围的规则而在任何地方定义和访问。

(4) 常量一旦定义就不能被重新定义或者取消定义。

(5) 常量的值只能是标量。

2.3 数据类型

PHP支持8种原始数据类型,其中标量类型4种,复合类型2种,特殊类型2种,如表2-3-1所示。实际上double和float是相同的,由于历史的原因,这两个名称同时存在。由于PHP是松散型语言,所以变量的类型不由程序员直接指定,而是根据该变量所处的上下文(或者说数据内容)在运行时决定。

表2-3-1 数据类型

标量类型	复合类型	特殊类型
boolean(布尔型) integer(整型) float(浮点型,也称double) string(字符串型)	array(数组) object(对象)	NULL Resource(资源)

2.3.1 布尔型

布尔(boolean)型变量只能有true和false两种取值,这两个值不区分大小写,分别对应逻辑"真"与逻辑"假"。下面是关于布尔型变量的范例。

【例2-3-1】 (代码位置:\2\testBool1.php)

```
<?php
  $a=true;
  $b=false;
  echo "下面打印两个布尔型变量的值<br/>";
  echo $b;
  echo "<br/>";
  echo $a;
?>
```

例中程序打印 true 值时显示的是 1，而打印 false 值时显示的是空，运行结果如下：

```
下面打印两个布尔型变量的值

1
```

布尔类型变量可用来控制流程走向，下面是布尔型变量控制流程的范例。

【例 2-3-2】 （代码位置：\2\testBool2.php）

```
<?php
  $sex=true;
  //==是一个操作符,它检测两个值是否相等,并返回一个布尔值
  if($sex==true){
    echo "是男同志<br/>";
  }
  $sex=false;
  if($sex==false){
    echo "是女同志";
}
?>
```

范例中 $sex 是布尔型变量，其初始值是 true，它控制第一个流程输出“是男同志”；然后变量取 false，它控制第二个流程输出“是女同志”。范例运行结果如下：

```
是男同志
是女同志
```

2.3.2 整型

PHP 中的整型指的是不包含小数部分的数值。在 32 位操作系统中，整型数据的有效范围是－2 147 483 648～＋2 147 483 647。整型数据可以用十进制（基数为 10）、八进制（基数为 8，以 0 作为前缀）和十六进制（基数为 16，以 0x 作为前缀）表示，并且可以包含正负符号。整型数据的表示方法如下面代码所示：

```
<?php
  $a=1234;          //十进制数
  $a=-123;          //负数
  $a=0123;          //八进制数（等于十进制 83）
  $a=0x1A;          //十六进制数（等于十进制 26）
?>
```

如果给定的数字超出了整型数据规定的范围，则会产生数据溢出。对于这种情况，PHP 会自动将整型数据转化为浮点型数据。下面是整型数据溢出的范例。

【例 2-3-3】 （代码位置：\2\testIntOverflow.php）

```
<?php
```

```
    $large_number=2147483647;
    var_dump($large_number);
    //输出 int(2147483647)
    $large_number=2147483648;
    var_dump($large_number);
    //输出 float(2147483648),因为超出整数的表示范围-2 147 483 648~+2147483647
    //同样也适用于十六进制表示的整数:从 2^31 到 2^32-1
    var_dump( 0xffffffff );
    //输出 float(4294967295),因为超出整数的表示范围-2 147 483 648~+2 147 483 647]
    //不适用于大于 2^32-1 的十六进制表示的数
    var_dump( 0x100000000 );
    //输出 int(2147483647)
    $million=1000000;
    $large_number=50000 * $million;
    var_dump($large_number);
    //输出 float(50000000000),因为超出整数的表示范围-2147483648~+2147483647
?>
```

2.3.3 浮点型

浮点型数据就是通常所说的实数,也称浮点数和双精度数,浮点数可以表示－1.8e308～＋1.8e308 之间的数据,并具有 14 位十进制数字的精度。浮点数可用于长度、重量和薪水等数据的表示。浮点型数据的表示方法如下面代码所示:

```
<?php
    $a=1.234;
    $b=1.2e3;
    $c=7E-10;              //科学表示法
?>
```

当浮点数超出范围时,会依据有效位数保留和取舍数值。下面是关于浮点数有效位数的范例。

【例 2-3-4】 (代码位置:\2\testFloat.php)

```
<?php
    $a=1234567890.1234567;
    echo $a;               //保留 14 位有效数字
?>
```

例中由于 1234567890.1234567 的长度超过 14 位,只保留 14 位有效数字,所以运行结果为 1234567890.1235。

2.3.4 字符串

字符串变量用于存储并处理文本片段。字符串是一个字符的序列。组成字符串的字符是任意的,可以是字母、数字或者符号。在 PHP 中没有对字符串的最大长度进行严

格的规定。在 PHP 中定义字符串常用两种方式定义：单引号和双引号。

1. 单引号

指定一个简单字符串的最简单的方法是用单引号(字符')括起来。单引号是字符串关键字,如果要打印这个关键字就需要用反斜线(\)转义表明其后的单引号只是普通字符而不是关键字。由于反斜线是转义关键字,如果要打印输出反斜线,需要用两个反斜线表示。如果在任何其他的字符前加了反斜线,那么反斜线将会被直接输出。下面是关于单引号和反斜线转义的范例。

【例 2-3-5】 (代码位置：\2\testSting1.php)

```
<?php
  echo 'Tom said: "I\'ll be back"<br/>';   //反斜线后是单引号,此处的单引号已不是字符串关
                                             键字
  echo 'You deleted C:\\*.*?<br/>';   //两个反斜线,第一个反斜线是转义关键字,第二个
                                        反斜线被转义后不再是转义关键字
  echo '将不会把 \n 转义为换行 <br/>';   //直接输出\n,因为紧接反斜线的字符既不是单引号
                                        也不是反斜线
  $a='abc';
  echo '变量名$a也不转义为变量$a的值 abc';//单引号里的$a不会被转义为变量$a
?>
```

程序运行结果如下：

```
Tom said: "I'll be back"
You deleted C:\*.*?
将不会把 \n 转义为换行
变量名$a也不转义为变量$a的值 abc
```

2. 双引号

如果用双引号(")括起字符串,那么 PHP 会对一些特殊的字符进行转义解析。表 2-3-2 是转义字符表。

表 2-3-2 转义字符

序　　列	含　　义
\n	换行
\r	回车
\t	制表符
\\	反斜线
\$	美元符号
\"	双引号
\[0-7]{1,3}	此正则表达式序列匹配一个用八进制符号表示的字符
\x[0-9A-Fa-f]{1,2}	此正则表达式序列匹配一个用十六进制符号表示的字符

注意：试图转义任何其他字符，反斜线本身也会被显示出来。双引号字符串最重要的一点是其中的变量名会被变量值替代。下面是双引号包含变量的范例。

【例 2-3-6】 （代码位置：\2\testTranMeaning.php）

```
<?
  $a=" abc ";
  echo "变量名$a转义为变量$a的值abc";      //输出结果：变量名abc转义为变量abc的值abc
?>
```

当用双引号指定字符串时，其中的变量会被解析。如果遇到$，PHP会尽可能多地取得后面的字符以组成一个合法的变量名。若要显示地指明变量名，用花括号把变量名括起来。应明确双引号中的变量名，以免发生变量解析错误。下面是解析双引号中含有变量字符串的范例。

【例 2-3-7】 （代码位置：\2\testString2.php）

```
<?php
  $beer='TsingTao';
  echo " $beer：中国啤酒名牌 <br/>";          //出错，这里用的是中文冒号
  echo " $beer ：中国啤酒名牌 <br/>";         //正常工作，这里用的是中文冒号，但在冒号
                                              //前加了个空格
  echo " $beer: 中国啤酒名牌 <br/>";          //正常工作，英文冒号：是无效变量标识符
  echo "$beer's taste is great <br/>";        //正常工作，单引号'是无效变量标识符
  echo "He drank some $beers <br/>";          //出错，'s' 是有效的变量标识符
  echo "He drank some ${beer}s <br/>";        //正常工作，使用花括号强制指明变量名
  echo "He drank some {$beer}s <br/>";        //正常工作，使用花括号强制指明变量名
?>
```

范例运行结果如下：

```
Notice: Undefined variable: beer：中国啤酒名牌 in C:\PHP\htdocs\test2_1.php on line 3

TsingTao ：中国啤酒名牌
TsingTao: 中国啤酒名牌
TsingTao's taste is great

Notice: Undefined variable: beers in C:\PHP\htdocs\test2_1.php on line 7
He drank some
He drank some TsingTaos
He drank some TsingTaos
```

2.3.5 数组

在使用PHP进行开发的过程中，经常需要创建许多相似的变量，这样的变量会使程序变得复杂，使用数组可以使程序变得简洁。数组就是一组数据的集合，把一系列数据组织起来，形成一个可操作的整体。PHP使用array关键字定义数组，使用数组名和下

标访问数组中的数据,有关数组的详细使用在后面的章节进行了详细的介绍。为了更好地认识数组的优势,下面给出两个存储数据的范例,这两个范例功能相同,不同的是,例2-3-8是用变量存取数据,而例2-3-9是用数组存取数据。

【例 2-3-8】 (代码位置:\2\testNonArrayCompare.php)

```
<?php
  $phpjc1='word';             //变量$phpjc1
  $phpjc2='excel';            //变量$phpjc2
  $phpjc3='outlook';          //变量$phpjc3
  $phpjc4='access';           //变量$phpjc4
  echo $phpjc1;
  echo "</br>";
  echo $phpjc2;
  echo "</br>";
  echo $phpjc3;
  echo "</br>";
  echo $phpjc4;
?>
```

【例 2-3-9】 (代码位置:\2\testArrayCompare.php)

```
<?php
  $phpjc=array('word','excel','outlook','access');     //定义一个数组
  for($i=0;$i<4;$i++)
    echo $phpjc[$i]."</br>";                            //元素下标为$i
?>
```

这两段程序功能一样,但前者使用变量而后者使用了数组,后者比前者简洁,相似变量数量越多,数组的优势就越明显。例2-3-9通过array关键字声明定义数组,数组名为phpjc,数组中有4个元素,这4个元素的值分别是word、excel、outlook和access,程序使用for循环输出数组元素。程序运行输出结果如下:

```
word
excel
outlook
access
```

2.3.6 类和对象

面向对象描述了对象以及对象相互之间的关系。对象是面向对象语言中的一个核心概念,对象就是类的一个实例。在面向对象语言中,人们把各个具体事物的共同特征抽取出来,形成一个一般的概念,也就构成了一个"类"。PHP中使用关键字class来定义一个类。类的命名一般使用首字符大写,而后每个单词首字符大写连接的方式以方便阅读。可以使用new关键字创建对象。在此,只给出类和对象的一个应用实例,后面章节

对类和对象进行了专门的讨论。

【例 2-3-10】 (代码位置:\2\testClassObj.php)

```
<?php
  class Person
  {
    public $name="张三";                    //定义 public 属性$name
    public $age=20;                          //定义 public 属性$age
    public function out()                    //定义 public 方法 out
    {
      echo " " . $this->name;                //输出属性$name
      echo " ";
      echo " " . $this->age;                 //输出属性$age
    }
  }
  $p=new Person();                           //创建对象
  $p->out();                                 //调用对象方法
?>
```

程序中定义了 Person 类,类中有 $ name、$ age 两个成员变量和 out 方法,程序创建了一个 Person 类型的对象并调用对象的方法 out,程序运行后会输出:张三 20。

2.3.7 NULL

NULL 是 PHP 中的一种特殊的数据类型,表示空值,即表示没有为变量设置任何值,null(空值)不区分大小写,null 和 NULL 是相同的。被赋空值可能有 3 种情况:变量未被赋予任何值;被赋空值 null;被函数 unset 处理过的变量。下面是这 3 种情况的范例。

【例 2-3-11】 (代码位置:\2\testNULL.php)

```
<?php
    $string1=null;                           //$string1 被赋空值
    $string3="str";                          //$string3 被赋值 str
    if(is_null($string1=null)){              //判断$string1 是否为空
        echo '变量($string1)直接赋值为 null ';
        echo 'string1=null';
    }
    echo '<p>';
    if(is_null($string2=null)){              //判断$string2 是否为空
        echo '变量($string2)没有被赋值 ';
        echo 'string2=null';
    }
    echo '<p>';
    unset($string3);
```

```
    if(is_null($string3=null)){                    //判断$string3是否为空
        echo '被 unset()释放过的变量($string3): ';
        echo 'string3=null';
    }
?>
```

程序中变量 $ string1 被直接赋空值，$ string2 没有定义，$ string3 先被赋值 str 之后又被函数 unset 注销掉该变量，程序运行结果如下：

```
变量($string1)直接赋值为 null string1=null
变量($string2)没有被赋值 string2=null
被 unset()释放过的变量($string3): string3=null
```

2.3.8 资源

资源是 PHP 提供的一种特殊数据类型，该数据类型用于表示一个 PHP 外部资源，比如一个数据库的访问操作，或者一个网络流的处理等。虽然资源也是一种数据类型，但是却不能直接对它进行操作。在 PHP 中，资源类型保存了到外部资源的一个引用，其实质就是为外部资源建立的特殊句柄，比如数据库连接、打开文件和图形画布区域等都会建立一个特殊句柄，程序通过这个句柄就可以访问 PHP 外部资源了。资源类型其实仅仅是一个整数，而内核可以根据这个整数值去一个类似资源池的地方寻找最终需要的数据。举个简单的文件访问例子：可以先使用 fopen 函数来打开一个文件获取句柄，之后只需把这个句柄传递给 feof、fread、fwrite 和 fclose 之类的函数，便可以对这个文件进行后续的读写等操作。PHP 提供了一些特定的函数，用于打开、使用和关闭资源。比如 mysql_connect 函数用于建立一个 MySQL 数据的连接，fopen 函数用于打开一个文件等。

下面是关于文件资源的应用范例，该范例由 text. txt 和 testResource1. php 两个文件组成，程序 testResource. php 会先打开文件 text. txt 获取资源句柄，然后通过该资源句柄去读取文件内容和打印内容，最后关闭资源。

【例 2-3-12】 （代码位置：\2\resource 目录）

文件 text. txt 内容如下：

```
0123456789
```

文件 testResource1. php 内容如下：

```
<?php
  $fp=fopen("test.txt", "rw");
  var_dump($fp);
  echo "<BR/>" ;
  $content=fread($fp,10);
  echo $content;
  fclose($fp);
```

```
?>
```

程序运行结果如下：

```
resource(3) of type (stream)
0123456789
```

数字3表示资源ID为3，stream表示资源类型名称，它是流类型，0123456789是资源的内容。程序通过函数fopen创建资源类型变量，该变量的值为整数3，该值作为资源ID和资源起着对应关系，函数var_dump用来输出资源变量$fp，函数fread通过资源ID读取资源的内容，函数fclose销毁资源变量。由于程序的外部资源类别众多，为了便于管理，PHP给外部资源提供多种不同的类型，这些类型有相应的创建者、使用者和销毁者等。PHP提供了数量众多的资源类型，表2-3-3只列出了常用的MySQL和FTP资源类型列表，创建者创建相应类型的资源，使用者使用该类型资源，销毁者用于关闭资源。

表 2-3-3 资源类型表

资源类型	创建者	使用者	销毁者
mysql link	mysql_connect 函数	mysql_query 等函数	mysql_close 函数
ftp	ftp_connect 函数	ftp_get 等函数	ftp_quit 函数

下面是MySQL数据库资源类型的一个应用范例，例中先使用mysql_connect创建数据库连接，如果连接成功则返回一个资源句柄给变量$con，然后函数var_dump通过该句柄打印数据库相关信息，最后函数mysql_close关闭数据库连接。有关数据库函数的使用参看后面章节。

【例 2-3-13】 （代码位置：\2\testDatabaseResource.php）

```
<?php
    $con=mysql_connect("localhost","root","111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    var_dump($con);
    mysql_close($con);
?>
```

2.4 运 算 符

运算符是指通过一个或多个表达式来产生另外一个值的某些符号，如“+”、“%”和“.”等都是运算符。在表达式2+1中，运算符“+”有两个操作数1和2。具有两个操作数的运算符叫做双目运算符。具有一个操作数的运算符叫做单目运算符，如表达式-6，

运算符“－”只有一个操作数 6，因此，这里的“－”是单目运算符。

2.4.1 运算符优先级

运算符有优先级，即表达式中多种运算符同时出现时，哪种运算符应该首先被应用。运算符优先级指定了两个表达式绑定得有多“紧密”。例如，表达式 1＋5 * 3 的结果是 16 而不是 18，因为乘号（“ * ”）的优先级比加号（“＋”）高。必要时可以用括号来强制改变优先级，例如，(1＋5) * 3 的值为 18。如果运算符优先级相同，其结合方向决定着运算应该从右向左求值，还是从左向右求值。表 2-4-1 从低到高列出了运算符的优先级，以及相同优先级的结合方向（同一行的运算符优先级相同，其按指定的结合方向进行运算）。

表 2-4-1　运算符优先级

结合方向	运　算　符
左	,
左	or
左	xor
左	and
右	=、+=、-=、*=、/=、.=、%=、&=、|=、^=、~=、<<=、>>=
左	?:
左	||
左	&&
左	|
左	^
左	&
无	==、!=、===、!==
左	<<、>>
左	+、-、.
左	*、/、%
右	!、~、++、--、(int)、(float)、(string)、(array)、(object)、@
右	[
无	new

注意，尽管!比＝的优先级高，PHP 仍旧允许类似这样的表达式：if (! $a=foo())，在此例中 foo()的输出被赋给了 $a。

2.4.2 算术运算符

PHP 的算术运算符有加(＋)、减(－)、乘(*)、除(/)和取模(%)和取反(－)。这些运算符的用法和学校里学到的数学知识一样。除号(/)总是返回浮点数，即使两个运算数是整数(或由字符串转换成的整数)也是这样。表 2-4-2 给出了算术运算符和实例。

表 2-4-2　算术运算符

算术形式	名　称	结　　果	实　例
$a + $b	加法	$a和$b的和	1+2=3
$a - $b	减法	$a和$b的差	1-2=-1
$a * $b	乘法	$a和$b的积	1*2=2
$a / $b	除法	$a除以$b的商	1/2=0.5
$a % $b	取模	$a除以$b的余数	1%2=1

2.4.3　赋值运算符

在PHP中,符号“=”并不是表示相等,而表示赋值。它的含义是将一个值指定给一个变量,更专业一点地说,它表示把符号“=”右边的表达式的值赋给左边的变量,如 $a=5表示将5赋给$a。赋值表达式的值也就是所赋的值,这就是说,$a=5的值是5。在基本赋值运算符之外,还有适合于所有二元算术和字符串运算符的“组和运算符”,这可让一个表达式中使用它的值并把表达式的结果赋给它,例如:

```
<?php
  $a=3;
  $a +=5;            //$a的值为8,它和$a=$a +5等价
  $b="Hello ";
  $b.="There!";      //$b的值为"Hello There!",它和$b=$b."There!"是等价的
?>
```

2.4.4　位运算符

位运算符允许对整型数中指定的位进行置位。如果左右参数都是字符串,则位运算符将操作这个字符串中的字符。

```
<?php
  echo 12 ^ 9;       //输出 '5'
?>
```

符号^表示按位异或运算,12和9的二进制值分别是1100和1001,两数按位异或运算后的二进制值是0101,即值5。表2-4-3给出了位运算符及其实例。

表 2-4-3　位运算符

位运算形式	名　　称	说　　明	实　　例
$a & $b	And(按位与)	将$a和$b的值按位进行与运算	1&2=0
$a | $b	Or(按位或)	将$a和$b的值按位进行或运算	1|2=3

续表

位运算形式	名 称	说 明	实 例
$a ^ $b	Xor(按位异或)	将$a和$b的值按位进行异或运算	4 ^ 5=1
~$a	Not(按位非)	将$a中为0的位设为1,为1的位设为0	~(4)=-5,4取反是-11,但负数用补码表示,对应的原码是-5
$a<<$b	Shift left(左移)	将$a中的位向左移动$b次(每一次移动都表示乘以2)	1<<2=4
$a>>$b	Shift right(右移)	将$a中的位向右移动$b次(每一次移动都表示除以2)	8>>1=4

2.4.5 比较运算符

比较运算符用来对两个值进行比较。表2-4-4列举了主要的比较运算符及其可能的运算结果。

表2-4-4 比较运算符

比较形式	名 称	结 果	实 例
$a==$b	等于	true,如果$a等于$b	$a="1";$b=1;$c=($a==$b);var_dump($c);输出:bool(true)
$a===$b	全等	true,如果$a等于$b,并且它们的类型也相同	$a="1";$b=1;$c=($a===$b);var_dump($c);输出:bool(false)
$a!=$b	不等	true,如果$a不等于$b	$a="1";$b=1;$c=($a!=$b);var_dump($c);输出:bool(false)
$a<>$b	不等	true,如果$a不等于$b	$a="1";$b=1;$c=($a<>$b);var_dump($c);输出:bool(false)
$a!==$b	非全等	true,如果$a不等于$b,或者它们的类型不同	$a="1";$b=1;$c=($a!==$b);var_dump($c);输出:bool(true)
$a<$b	小于	true,如果$a严格小于$b	$a="1";$b=1;$c=($a<$b);var_dump($c);输出:bool(false)
$a>$b	大于	true,如果$a严格大于$b	$a="1";$b=1;$c=($a>$b);var_dump($c);输出:bool(false)
$a<=$b	小于等于	true,如果$a小于或者等于$b	$a="1";$b=1;$c=($a<=$b);var_dump($c);输出:bool(true)
$a>=$b	大于等于	true,如果$a大于或者等于$b	$a="1";$b=1;$c=($a>=$b);var_dump($c);输出:bool(true)

2.4.6 递增和递减运算符

PHP有和C语言风格相同的递增/递减运算符。递增是指对当前表达式的值增加1,递减正相反,对表达式的值减1。本书仅限整数表达式的递增/递减运算,表2-4-5是递

增和递减运算符。

表 2-4-5　递增和递减运算符

加减形式	名称	说　　明	实　　例
++$a	前加	$a的值加1,然后返回$a	$a=10; $b=++$a; 输出: $b=11,$a=11
$a++	后加	返回$a,然后将$a的值加1	$a=10; $b=$a++; 输出: $b=10,$a=11
--$a	前减	$a的值减1,然后返回$a	$a=10; $b=--$a; 输出: $b=9,$a=9
$a--	后减	返回$a,然后将$a的值减1	$a=10; $b=$a--; 输出: $b=10,$a=9

2.4.7　逻辑运算符

逻辑运算符用来对布尔值进行运算,包括与、或、非和异或运算。与和或均各有两种不同形式的运算符的原因是它们操作的优先级不同。表2-4-6给出了逻辑运算符和实例。

表 2-4-6　逻辑运算符和实例

逻辑形式	名　称	结　　果	实　　例
$a and $b	And(逻辑与)	true,如果$a与$b都为true	$a=true; $b=false; $c=($a and $b); var_dump($c); 输出: bool(false)
$a or $b	Or(逻辑或)	true,如果$a或$b任一为true	$a=true; $b=false; $c=($a or $b); var_dump($c); 输出: bool(true)
$a xor $b	Xor(逻辑异或)	true,如果$a和$b不相同	$a=true; $b=false; $c=($a xor $b); var_dump($c); 输出: bool(true)
!$a	Not(逻辑非)	true,如果$a不为true	$a=true; $c=(!$a); var_dump($c); 输出: bool(false)
$a && $b	And(逻辑与)	true,如果$a与$b都为true	$a=true; $b=false; $c=($a && $b); var_dump($c); 输出: bool(false)
$a || $b	Or(逻辑或)	true,如果$a或$b任一为true	$a=true; $b=false; $c=($a || $b); var_dump($c); 输出: bool(true)

2.4.8　字符串运算符

字符串运算符将两个字符串连接成一个新的字符串。有两个字符串运算符。第一个是连接运算符("."),它返回其左右参数连接后的字符串。第二个是连接赋值运算符

(“.＝”),它将右边参数附加到左边的参数后。下面是字符串连接运算符的范例。

【例 2-4-1】 (代码位置：\2\testStrOp.php)

```
<?php
  $a="Hello ";
  $b=$a . "World!";
  $a="Hello ";
  $a .="World!";
  echo $b;                    //输出 Hello World!
  echo $a;                    //输出 Hello World!
?>
```

2.4.9 其他运算符

PHP 支持一个错误控制运算符：@。当将其放置在一个 PHP 表达式之前，该表达式可能产生的任何错误信息都被忽略掉。下面是错误控制运算符的范例。

【例 2-4-2】 (代码位置：\2\testErrorOp.php)

```
<?php
  $a=0;
  $b=11;
  $c=@($b/$a);                //分母为零出错,但@运算符忽略该错误
  echo $c;                    //没有输出
?>
```

PHP 支持一个执行运算符：反引号(`)。注意这不是单引号“'”!PHP 将尝试将反引号中的内容作为外壳命令来执行，并返回命令的输出信息，例如，可以赋给一个变量而不是简单地丢弃到标准输出。使用反引号运算符“`”的效果与函数 shell_exec 相同(函数在 Linux 环境有效)。下面是执行运算符的范例。

注意：范例需在 Linux 环境下运行，因为命令“ls －al”是 Linux 命令，如需在 Windows 下运行，可把命令改为 DOS 命令，例如 dir。

【例 2-4-3】 (代码位置：\2\testExec.php)

```
<?php
  $output=`ls -al`;         //执行 Linux 环境下的 ls 命令,并把命令的结果返回给变量$output
  echo "<pre>$output</pre>";
?>
```

2.5 控制结构

所有的 PHP 程序都由语句构成，程序就是一系列语句的序列。计算机通过执行这些语句可以完成特定的功能。一般情况下，程序都是从第一条语句开始执行，按顺序执

行到最后一句。但有时因为某种情况,需要改变程序的执行顺序,这就需要对程序的流程进行控制。PHP 通过控制语句对程序的执行走向进行控制。PHP 主要有 3 种控制结构:顺序结构、选择结构和循环结构。顺序结构使程序从第一条语句开始,按顺序执行到最后一句。在选择结构中,程序可以根据某个条件是否成立,选择执行不同的语句。在循环结构中,可以使程序根据某种条件和指定的次数,使某些语句执行多次。PHP 控制语句根据控制结构可分为下面 3 类语句。

(1) 选择语句:if/if…else、switch。

(2) 循环语句:foreach、while、do while、for。

(3) 跳转语句:break、continue。

2.5.1 if/if…else 语句

if 条件语句又称分支语句,if 语句是包括 PHP 在内的众多编程语言最重要的特性之一,它允许按照条件执行代码片段。PHP 的 if 结构和 C 语言相似,其语法结构如下所示:

```
if ( 表达式 ) {
  如果表达式的值为 true 执行的语句;
}
```

if 后面小括号中的“表达式”就是执行的条件,条件只能是布尔型值,其通常是由比较运算符或者逻辑运算符组成的表达式所计算的结果值,或是一些返回布尔型值的函数等。如果是传入其他类型的值也会自动转换为布尔型的 true 或 false。如果“表达式”为 true,则执行花括号中的语句,否则不执行。不论结果如何,接下来都将执行 if 后面的语句。其过程可以表示成图 2-5-1。例 2-5-1 是 if 语句范例。

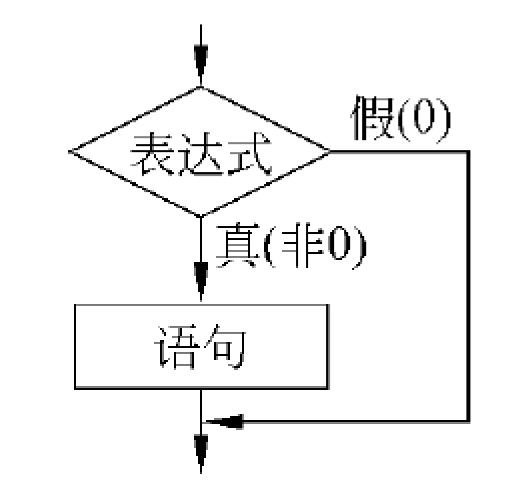


图 2-5-1 if 语句流程

【例 2-5-1】 (代码位置:\2\testIf. php)

```
<?php
  $age=21;
  if($age>20){
  echo '正在上大学';
  }
?>
```

例中因变量 $age 值大于 20,所以会执行 if 语句花括号中的语句,它会在浏览器中输出“正在上大学”。

if…else 语句是双向条件选择语句,经常用在满足某个条件时执行一个语句块,而在不满足该条件时执行其他语句块。else 延伸了 if 语句,可以在 if 语句中的表达式的值为 false 时执行语句块。要注意的是,else 语句是 if 语句的从句,必须和 if 一起使用,不能单独存在。if…else 语句的格式如下:

```
if(表达式 ){
```

```
  语句块 1;            //如果表达式的值为 true 执行的语句
}else{
  语句块 2;            //如果表达式的值为 false 执行的语句
}
```

在上面的格式中，如果表达式的值为真，则执行 if 后面的语句块 1；如果表达式值为假，则执行 else 后面的语句块 2。语句块 1 和语句块 2 可以是单条语句，也可以是复合语句，如果是复合语句必须使用花括号"{}"括起来。if…else 流程可用图 2-5-2 表示。例 2-5-2 是 if…else 的范例。

图 2-5-2 if…else 流程

【例 2-5-2】 （代码位置：\2\testIfElse.php）

```
<?php
  $age=18;
  if($age>20){
    echo'正在上大学';
  }else{
    echo'还没有上大学';
  }
?>
```

程序会在浏览器中输出"还没有上大学"。

2.5.2 switch 分支结构

if 语句主要处理有两个分支的情况，但如果分支较多，则 if 语句嵌套的层数就太多了，程序不但冗长，而且可读性差。因此 PHP 语言又提供了一种专门用于处理多分支结构的条件选择语句，称为 switch 语句，又称为开关语句。很多场合下需要把一个变量（或表达式）与很多不同的值比较，并根据它等于哪个值来执行不同的代码块，switch 语句就可以很好地应用在这种场合。switch 语句由一个控制表达式和多个 case 标签组成，case 标签后紧跟一个代码块，case 标签作为这个代码块的标识。switch 语句的语法格式如下：

```
switch(表达式 ){
case 值 1:
  符合值 1 的执行语句;
  break;
case 值 2:
  符合值 2 的执行语句;
  break;
…
case 值 n:
  符合值 n 的执行语句;
  break;
```

```
default:
  所有值都不符合的执行语句
}
```

switch 语句的执行过程如下。

(1) 计算 switch 语句后面的表达式的值,当表达式的值与某一个 case 后面的常量的值相等时,就执行此 case 后面的语句,若所有的 case 中的常量的值都没有与表达式的值相匹配的,就执行 default 后面的语句,当没有 default 语句时,则什么也不执行。

(2) 执行完一个 case 后面的语句后,程序执行的流程转移到下一个 case 继续执行。"case 值 *n*"只是起语句标号作用,并不是在该处进行条件判断。在执行 switch 语句时,根据 switch 后面表达式的值找到匹配的入口标号,就从此标号开始执行下去,不再进行判断。只有当遇到 break 语句或执行完全部 switch 内的语句时,才跳出 switch 语句。

下面是 switch 分支语句的范例。

【例 2-5-3】 (代码位置:\2\testSwitch.php)

```
<?php
$i=1;
switch($i){
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
    default:
        echo "i is not equal to 0, 1 or 2";
}
?>
```

例中如果 $i 等于 0,浏览器将输出"i equals 0";如果 $i 等于 1,浏览器将输出"i equals 1";如果 $i 等于 2,浏览器将输出"i equals 2";如果 $i 都不等于 0、1 和 2,程序会输出"i is not equal to 0, 1 or 2"。由于程序中 $i 值为 1,所以程序会输出"i equals 1"。

2.5.3 while 循环结构

while 循环是 PHP 中最简单的循环类型,其语法基本格式如下:

```
while(表达式){
  循环语句;
}
```

while 语句的含意很简单,它告诉 PHP 程序只要 while 表达式的值为 true 就重复执

行循环语句。表达式的值在每次开始循环时检查,所以即使这个值在循环语句中改变了,语句也不会停止执行,直到本次循环结束。有时候如果 while 表达式的值一开始就是 false,则循环语句一次都不会执行。其执行流程如图 2-5-3 所示。

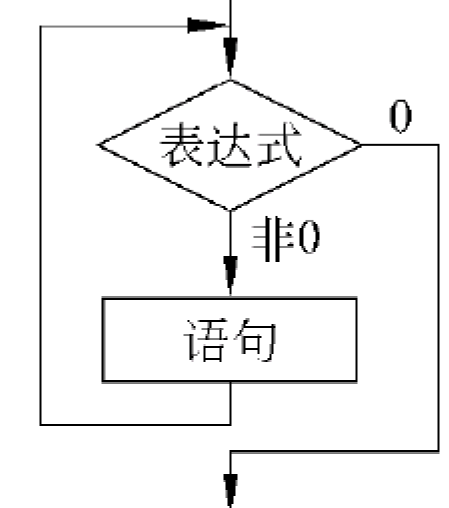


图 2-5-3 while 流程

下面是 while 循环语句的范例。

【例 2-5-4】 (代码位置:\2\testWhile.php)

```
<?php
  $int=1;
  while($int<=10){
    echo"这是执行第{$int}次循环输出的结果。"."<br>";
    $int++;
  }
?>
```

例中程序会循环执行 10 次,每次循环 $int 都会自加,当 $int 值为 11 时将结束 while 循环,其输出如下:

```
这是执行第 1 次循环输出的结果。
这是执行第 2 次循环输出的结果。
这是执行第 3 次循环输出的结果。
这是执行第 4 次循环输出的结果。
这是执行第 5 次循环输出的结果。
这是执行第 6 次循环输出的结果。
这是执行第 7 次循环输出的结果。
这是执行第 8 次循环输出的结果。
这是执行第 9 次循环输出的结果。
这是执行第 10 次循环输出的结果
```

2.5.4 do…while 循环结构

do…while 和 while 循环非常相似,区别在于表达式的值是在每次循环结束时检查而不是开始时检查,do…while 循环语句保证至少执行一次循环体语句,而 while 循环就不一定了(对于 while 语句,表达式真值在循环开始时检查,如果一开始就为 false 则整个循环立即终止)。do…while 语句的应用格式如下:

```
do{
  执行语句;
}while(表达式);
```

do…while 的执行过程如图 2-5-4 所示。

下面是 do…while 循环语句的范例。

【例 2-5-5】 (代码位置:\2\testDoWhile.php)

```
<?php
```

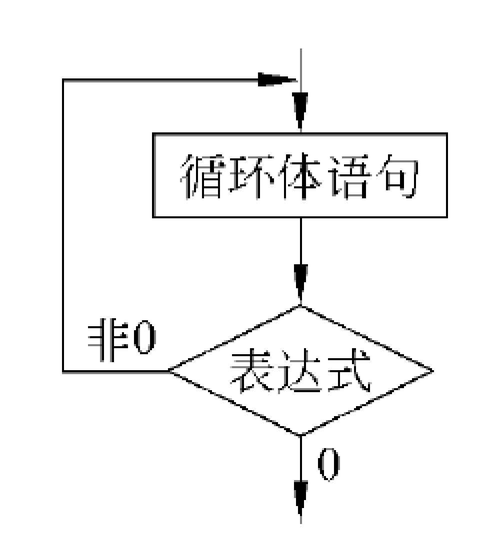


图 2-5-4 do…while 的执行过程

```
  $i=0;
  do {
    echo $i;
  } while($i >0);
?>
```

程序会先执行循环体语句输出变量 $i 的值,然后判断 $i 值是否大于 0,因为 $i 值为 0,表达式值为假,因此程序会跳出循环。程序输出结果为 0。

2.5.5 for 循环结构

while 循环通常用于条件型循环,即遇到特定的条件才停止循环。for 循环语句适用于明确知道重复执行次数的情况,for 语句将循环次数的变量预先定义好。虽然 for 语句是 PHP 中最复杂的循环结构,但用于以计数方式控制循环时非常方便。for 语句的应用格式如下:

```
for(初始化表示式;循环条件表达式;递增/递减规则表达式){
  循环语句;
}
```

for 语句有 3 个表达式:初始化表达式、循环条件表达式和递增/递减规则表达式,表达式间用分号隔开。初始化表达式用来给循环控制变量赋初值,它在循环开始前无条件求值一次。循环条件表达式是一个关系表达式,它决定什么时候退出循环,如果值为 true,则继续循环执行嵌套的循环语句;如果值为 false,则终止循环。递增/递减规则表达式在每次循环之后会对控制变量求值。当初始值比循环条件中的值小时要用递增规则计算控制变量,初始值比循环条件中的值大时要用递减规则计算控制变量,否则会出现死循环。for 语句执行过程可用图 2-5-5 表示。

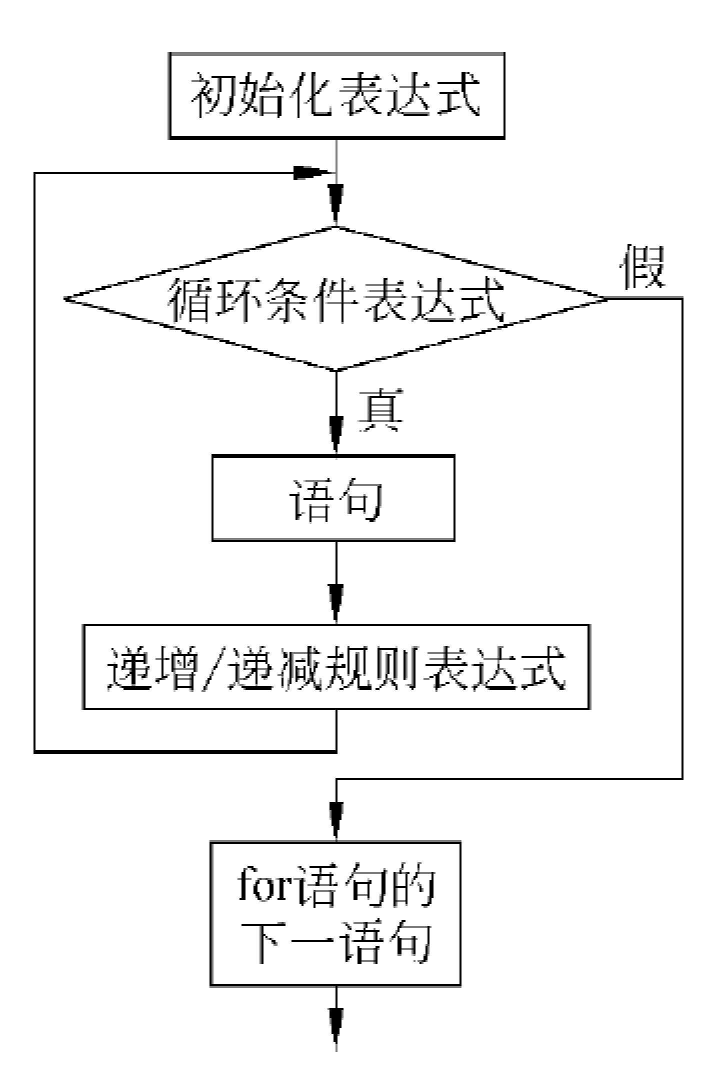


图 2-5-5 for 语句的执行过程

下面是 for 语句的范例。

【例 2-5-6】 (代码位置:\2\testFor1.php)

```
<?php
  for($int=1;$int<=10;$int++)
  {
    echo"这是执行第{$int}次循环输出的结果。"."<br>";
  }
?>
```

程序执行过程如下:

(1) 执行 $int=1 语句。

(2) 执行语句 $int<=10 判断是否小于 10,小于或等于 10 则继续执行花括号“{}”

里的语句,否则退出 for 循环继续执行 for 语句的下一条语句。

(3) 执行语句$int++让变量$int 值自加1,然后跳到(2)。

浏览器输出如下所示。

```
这是执行第 1 次循环输出的结果。
这是执行第 2 次循环输出的结果。
这是执行第 3 次循环输出的结果。
这是执行第 4 次循环输出的结果。
这是执行第 5 次循环输出的结果。
这是执行第 6 次循环输出的结果。
这是执行第 7 次循环输出的结果。
这是执行第 8 次循环输出的结果。
这是执行第 9 次循环输出的结果。
这是执行第 10 次循环输出的结果。
```

下面是 for 嵌套循环的范例。

【例 2-5-7】 (代码位置:\2\testFor2.php)

```
<?php
  for($i=1;$i<=5;$i++){
    for($j=1;$j<=$i;$j++){                //嵌套使用 for 语句
      echo "$i × $j=".$i * $j."  ";
    }
    echo"<br>";
  }
?>
```

程序输出结果如下:

```
1×1=1
2×1=2 2×2=4
3×1=3 3×2=6 3×3=9
4×1=4 4×2=8 4×3=12 4×4=16
5×1=5 5×2=10 5×3=15 5×4=20 5×5=25
```

2.5.6 foreach 循环结构

foreach 也是循环语句。foreach 仅能用于数组,当试图将其用于其他数据类型或者一个未初始化的变量时会产生错误。foreach 有两种语法格式,第一种格式如下:

```
foreach(数组名称 as $value){
  执行语句;
}
```

这种方式在遍历数组的时候,只是将数组元素的值赋给定义的变量 value,其中变量 value 的名称是可以任意命名的,这种方式输出的是变量 value 的值,变量 key 的值不会

输出。下面是关于 foreach 语句第一种格式的范例。

【例 2-5-8】（代码位置：\2\testForeach1.php）

```
<?php
  $number=array('1'=>'数学','2'=>'语文','3'=>'英语');
  foreach($number as $value){
    echo$value."、";
  }
?>
```

程序输出如下：

数学、语文、英语。

foreach 语句的第二种格式如下：

```
foreach(数组名称 as$key=>$value){
  执行语句;
}
```

这种方式除了将数组元素的值赋给变量 value 之外，还将各数组元素的键赋给变量 key。在这里变量 key 和变量 value 的名称是可以任意命名的。下面是 foreach 语句第二种格式的范例。

【例 2-5-9】（代码位置：\2\testForeach2.php）

```
<?php
  $number=array('数学','语文','英语','物理','化学');
  foreach($number as $key=>$value){
  echo "$key=>$value<br/>";
  }
?>
```

程序运行后输出如下：

```
0=>数学
1=>语文
2=>英语
3=>物理
4=>化学
```

第 3 章



PHP 数组

3.1 什么是数组

生活中,一个笔记本电脑包只能放一台笔记本电脑,一个书包可以放多本书,一个抽屉可存放各种不同类型的物体。电脑包、书包和抽屉等都是可以容纳物体的容器。在 PHP 中,一个变量可以理解为只能存储单个值的容器,而一个数组可以理解为能存储多个值的容器。数组就像书包和抽屉一样,可以存储多个相同或不同类型的物体。数组是数据值的集合,是一个可以存储一组或一系列数值的变量。存储在数组中的值称为数组元素。一个数组可以具有许多个元素。PHP 允许一个数组可以包括不同数据类型的元素,例如字符串、数字或另一个数组。一个包含其他数组的数组称为多维数组,一个只包含值的数组称为一维数组。数组中的元素是在索引(也称下标)的基础上被引用和操作的。根据索引类型的不同,PHP 数组可分为数字索引数组和关联数组。数字索引数组的索引是从 0 开始的整数,而关联数组的索引是字符串。实质上,在 PHP 内部所有的数组都被存储为关联数组,只是数字索引数组将索引特别指定到了数字。

注意:关联数组与数字索引数组唯一的区别在于索引的类型。与其他许多语言中的数组不同,PHP 不需要在创建数组时指定数组的大小,甚至不需要在使用数组前先行声明。

3.2 数组的类型

3.2.1 数字索引数组

大多数编程语言都支持数字索引数组。数字索引数组是指数组元素的索引为整数的数组,索引从 0 开始,依次递增。当需要通过相对位置来确定数组元素时,通常使用数字索引数组。数字索引数组也称为枚举数组、标量数组或简单数组,表 3-2-1 是一个数字索引数组的示例。

表 3-2-1 索引数组 $ name

数组元素	$ name[0]	$ name[1]	$ name[2]	$ name[3]	$ name[4]
元素值	张明	李四	王小五	赵勤	李小平

表中有一个索引数组,数组名为$name,数组含有5个数组元素,这5个数组元素的值是"张明"、"李四"、"王小五"、"赵勤"和"李小平"。可以通过索引(也称下标)0来引用"张明",通过索引1来引用"李四",通过索引2来引用"王小五",依此类推。要引用PHP中的元素,需要使用一对中括号([]),例如显示索引2的数组元素值的语句如下:

```
echo $name[2]
```

注意:PHP的数字索引数组默认从0开始,而不是1,这与C、C++和Java等许多编程语言相同,数组最后一个索引号等于数组元素的个数减1。

3.2.2 关联数组

数字索引数组在各种编程语言中非常常见。但对于关联数组,只有使用过PHP或Perl的程序员才不会感到陌生。关联数组允许数组中每个元素除了可以使用数字索引外,还可以使用字符串作为索引。如果需要通过名称访问数组中的元素,那么使用字符串作为数组的索引将是非常方便的。使用字符串作为索引的数组即称为关联数组。表3-2-2是一个关联数组的示例。

表3-2-2 关联数组$name

数组元素	$name["user1"]	$name["user2"]	$name["user3"]	$name["user4"]	$name["user5"]
元素值	张明	李四	王小五	赵勤	李小平

表中有一个关联数组,数组名为$name,数组含有5个数组元素,这5个数组元素的值是"张明"、"李四"、"王小五"、"赵勤"和"李小平"。其索引值分别为字符串user1、user2、user3、user4和user5。通过字符串索引值可以访问数组元素,比如显示索引为user2的数组元素的语句如下:

```
echo $name[user2]
```

注意:使用数字索引数组和关联数组的区别之一在于关联数组以字符串为索引,不能通过简单的加1或减1直接计算出数组中的下一个或前一个的有效索引值。

3.3 数组的创建和初始化

3.3.1 一维索引数组

在PHP中声明数组的方式主要有两种:一是应用函数array声明数组;一是直接为数组元素赋值。函数array用来新建一个数组。它接受一定数量用逗号分隔的key => value参数对。使用array函数创建数组的格式如下:

```
array([key=> ] value, ...)
```

key是索引,又叫键值或者下标,它可以是数字或者字符串,value可以是任何值。下

例是使用 array 创建下标为字符串的数组范例。

【例 3-3-1】 （代码位置：\3\testIndexArray1. php）

```
<?php
  $myarray=array("1"=>"编","2"=>"程","3"=>"词","4"=>"典");
  print_r($myarray)
?>
```

程序运行后浏览器输出如下：

```
Array([1]=>编 [2]=>程 [3]=>词 [4]=>典)
```

上例使用函数 array 创建一个包含 4 个元素的数组，元素的下标分别为 1、2、3 和 4，同时，该函数还给元素赋初值，初值分别是“编”、“程”、“词”和“典”。下面是创建下标为整数的数组范例。

【例 3-3-2】 （代码位置：\3\testIndexArray2. php）

```
<?php
  $phpjc=array(
    0=>'word',                    //指定下标 0
    3=>'excel',                   //指定非连续的下标 3
    'outlook',                    //未指定下标
    'access');                    //创建一个数组,并初始化数组元素
  print_r($phpjc);
?>
```

输出结果如下：

```
Array([0]=>word [3]=>excel [4]=>outlook [5]=>access)
```

上例定义了一个数组，名字叫 phpjc，第一个元素（即 $ phpjc[0]）的值是 word，第二个元素（即 $ phpjc[3]）值是 excel、outlook 和 access 分别是第三和第四个元素的值。值得注意的是，此处数组元素的下标并不是连续的，$ phpjc[1]和 $ phpjc[2]并未定义。如果数组元素不指定下标，则按递增的方式生成下标。此处尽管未指定下标 4 和 5，但第三个元素 $ phpjc[4]和第四个元素 $ phpjc[5]的下标会在第二个元素 $ phpjc[3]的下标 3 基础上依次递增。

如果在创建数组时不知所创建数组的大小，或在实际编写程序时数组的大小可能发生变化，可以通过给变量赋予一个没有参数的 array()来创建空数组，然后可以通过使用方括号[]语法来添加值。下面是直接为数组元素赋值的范例。

【例 3-3-3】 （代码位置：\3\testIndexArray3. php）

```
<?php
  $phpjc=array();                 //创建一个数组
  $phpjc[0]="one";                //直接为数组元素赋值
  $phpjc[1]="two";                //直接为数组元素赋值
  echo $phpjc[0]."<br>";
```

```
  echo $phpjc[1];
?>
```

上例先创建了一个数组，数组名为 phpjc，然后通过赋值运算符给数组元素赋值，最后输出数组元素的值。运行后浏览器输出结果如下：

```
one
two
```

下面是使用赋值的方式初始化数字索引数组的例子：

```
$name=array();
$name[0]="张明";
$name[1]="李四";
$name[2]="王小五";
$name[3]="赵勤";
Sname[4]="李小平";
```

当索引值是递增的，可以不在方括号内指定索引值，默认的索引值从 0 开始依次增加，即为 0、1、2、3 和 4 等，具体代码如下：

```
$name=array();
$name[]="张明";
$name[]="李四";
$name[]="王小五";
$name[]="赵勤";
Sname[]="李小平";
```

这种不显式地指定索引值的赋值方法，也称简单赋值方法，在初始化索引值连续递增的数组时非常简便。对于非连续索引的数组，对其初始化可以直接指定索引值，具体代码如下：

```
$name=array();
$name[0]="张明";
$name[10]="李四";
$name[20]="王小五";
$name[30]="赵勤";
$name[40]="李小平";
```

其中数组 name 有 5 个数组元素，其索引值分别为 0、10、20、30 和 40。它们对应的元素值分别为“张明”、“李四”、“王小五”、“赵勤”和“李小平”。

3.3.2 一维关联数组

关联数组的初始化可以采用与数字索引数组类似的方法，通过直接为数组元素赋值来完成。下面是以赋值的形式初始化关联数组的范例。

【例 3-3-4】 （代码位置：\3\testAssociateArray1. php）

```
<?php
  $name=array();
  $name['user1']="张明";
  $name['user2']="李四";
  $name['user3']="王小五";
  $name['user4']="赵勤";
  $name['user5']="李小平";
  print_r($name);
?>
```

程序运行输出结果如下：

```
Array([user1]=>张明 [user2]=>李四 [user3]=>王小五 [user4]=>赵勤 [user5]=>李小平)
```

使用 array 函数来完成关联数组的初始化也是常用的一种方法，其中索引值与数组元素值之间需要使用运算符“=>”。下面是使用 array 初始化数组 $score 的范例。

【例 3-3-5】 （代码位置：\3\testAssociateArray2. php）

```
<?php
  $score=array("张明"=>80,
        "李亮"=>90,
        "王平"=>67,
        "张晓红"=>77,
        "赵高"=>87);
  print_r($score);
?>
```

程序输出结果如下：

```
Array([张明]=>80 [李亮]=>90 [王平]=>67 [张晓红]=>77 [赵高]=>87)
```

上例程序等同于如下代码。

【例 3-3-6】 （代码位置：\3\testAssociateArray3. php）

```
<?php
  $score=array();
  $score["张明"]=80;
  $score["李亮"]=90;
  $score["王平"]=67;
  $score["张晓红"]=77;
  $score["赵高"]=87;
  print_r($score);
?>
```

3.3.3 多维数组

如果数组 A 的元素仍为数组，那么数组 A 就构成了多维数组。一个包含数组的数

组称为二维数组,如果二维数组中的数组元素仍包含数组,就构成了一个三维数组,依此类推,可以创建四维数组、五维数组等多维数组。但最常见的是二维数组。可以这样理解多维数组,一个大容器里含有很多小容器,小容器里还可以再含有很多更小的容器,依此类推。为简单起见,多维数组主要介绍二维数组。可以把二维数组想像成一个矩阵,或者是一个具有宽度和高度或者行和列的表格。

下面是二维数组的范例。表 3-3-1 是成绩表,该成绩表含有 3 条记录,其中每一行表示一个学生记录,每一列分别代表学生的“姓名”、“学号”、“性别”和“成绩”。例中使用一个 3×4 的二维数组 info 表示该成绩表,3 表示数组 info 有 3 个元素,4 表示数组 info 中的每个元素都是一个含有 4 个元素的数组。

表 3-3-1 成绩表

姓 名	学 号	性 别	成 绩
张三	A01	男	90
李四	A02	男	12
老五	B02	男	80

【例 3-3-7】 (代码位置:\3\testMDimArray.php)

```
<?php
  $info=array(array("张三","A01","男","90"),
              array("李四","A02","男","12"),
              array("老五","B02","男","95"));
  print_r($info);
?>
```

程序输出结果如下:

```
Array([0]=>Array([0]=>张三 [1]=>A01 [2]=>男 [3]=>90 )[1]=>Array([0]=>李四
[1]=>A02 [2]=>男 [3]=>12 )[2]=>Array([0]=>老五 [1]=>B02 [2]=>男 [3]=>95 ))
```

数组 info 第一个元素:[0]=>Array([0]=>张三[1]=>A01[2]=>男[3]=>90)。
数组 info 第二个元素:[1]=>Array([0]=>李四[1]=>A02[2]=>男[3]=>12)。
数组 info 第三个元素:[2]=>Array([0]=>老五[1]=>B02[2]=>男[3]=>95)。
上例程序等同于下面的代码。

【例 3-3-8】 (代码位置:\3\testMDimArray1.php)

```
<?php
  $info=array();
  $info[0]=array("张三","A01","男","90");
  $info[1]=array("李四","A02","男","12");
  $info[2]=array("老五","B01","男","90");
  print_r($info);
?>
```

3.4 数组的遍历

3.4.1 一维数组的遍历

一般情况下，遍历一个数组有 for、while 和 foreach 三种方法。其中最简单方便的是 foreach。对于遍历同样一个数组，foreach 速度最快，最慢的则是 while。从原理上来看，foreach 是对数组副本进行操作（通过复制数组），而 while 则通过移动数组内部指标进行操作，按照一般逻辑会认为 while 应该比 foreach 快，因为 foreach 在开始执行的时候首先把数组复制进去，而 while 直接移动内部指标。但结果刚刚相反，原因是 foreach 是 PHP 内部实现循环而 while 是通用的循环结构。在应用中更常用 foreach 遍历数组。在 PHP 5 中，foreach 还可以遍历类的属性。使用 foreach 的不足是只能顺序读取数组元素而不能随机读取。使用 for 语句可以随机读取数组元素，但需要计算数组长度，并且不能读取键。使用 while 语句通常需要其他函数配合。下面是使用 foreach 遍历数组的范例。

【例 3-4-1】 （代码位置：\3\testArrayForeachTravel1.php）

```
<?php
  $colors=array('red','blue','green','yellow');
  foreach($colors as $color){
    echo "Do you like $color? <br />";
  }
?>
```

程序每次遍历数组 colors 时，数组中的每个元素都用变量 $color 表示，运行结果显示如下：

```
Do you like red?
Do you like blue?
Do you like green?
Do you like yellow?
```

上例遍历数组时并不能知道数组元素的下标。下面是使用 foreach 的另一种形式遍历数组的范例，该范例可以知道每个元素的下标。

【例 3-4-2】 （代码位置：\3\testArrayForeachTravel2.php）

```
<?php
  $colors=array('red','blue','green','yellow');
  foreach($colors as $key=>$color){
    echo "$key :Do you like $color? <br />";
}
?>
```

程序每次遍历数组 colors 时，数组元素的键用变量 $key 表示，元素的值用变量 $color 表示，运行结果显示如下：

```
0 :Do you like red?
1 :Do you like blue?
2 :Do you like green?
3 :Do you like yellow?
```

下面是使用 for 遍历数组的范例。

【例 3-4-3】 （代码位置：\3\testArrayForTravel. php）

```
<?php
$arr=array("0"=>"zero","1"=>"one","2"=>"two");
for($i=0;$i <count($arr);$i++){
    $str=$arr[$i];
    echo "the number is $str.<br />";
}
?>
```

例中需要使用函数 count 计算数组 $ arr 的长度，程序运行结果如下：

```
the number is zero.
the number is one.
the number is two.
```

下例是使用 while 遍历数组的范例，其需要使用函数 each 和 list 配合 while 才能遍历数组。

【例 3-4-4】 （代码位置：\3\testArrayWhileTravel. php）

```
<?php
  $colors=array('red','blue','green','yellow');
  while(list($key,$val)=each($colors)){
    echo "Other list of $val.<br />";
  }
?>
```

例中每次循环会通过函数 each 取出数组 $ colors 中的下一个元素，函数 list 的作用是把数组元素的键值和值分别赋给变量 $ key 和 $ val。程序运行结果如下：

```
Other list of red.
Other list of blue.
Other list of green.
Other list of yellow.
```

3.4.2 多维数组的遍历

多维数组和一维数组的遍历类似，下面是多维数组遍历的范例。

【例 3-4-5】 （代码位置：\3\testMDimArrayForeachTravel1. php）

```
<?PHP
  //定义一个多维数组
```

```
  $more=array(
    "numbers"=>array(1,2,3,4,5,6,7),
    "colors"=>array("红","蓝","绿","黄","紫","青","橙")
  );
  //遍历一个多维数组
  foreach($more as $MYARRAY){
    foreach($MYARRAY as $key=>$value){
      echo $value;
    }
  }
?>
```

程序运行结果如下：

```
1234567红蓝绿黄紫青橙
```

值得注意的是，用这种方法遍历多维数组时，多维数组的低维元素应该是数组而不是值。如果多维数组中的元素含有数组和数值，则可以通过 is_array 函数判断元素是否是数组，并根据情况进行深度遍历，下面给出相应范例。

【例 3-4-6】 （代码位置：\3\testMDimArrayForeachTravel2.php）

```
<?PHP
  //定义一个多维数组
  $more=array(
    "numbers"=>10,
    "colors"=>array("红","蓝","绿","黄","紫","青","橙")
  );
  //遍历一个多维数组
  foreach($more as $MYARRAY){
    if(is_array($MYARRAY))
      foreach($MYARRAY as $key=>$value){
        echo $value;
      }
    else echo $MYARRAY;
  }
?>
```

程序运行结果如下：

```
10红蓝绿黄紫青橙
```

3.5 数组函数库

PHP 提供专门用于数组的函数，这些函数作为 PHP 核心的一部分，无须安装即可使用。这些函数允许用多种方法来操作数组和与之交互。数组的本质是储存、管理和操作

一组变量。下面介绍排序、逆排序和删除等函数,其他函数请参看其他相关资料。

3.5.1 排序

可用函数 sort 对数组中的元素进行排序。其格式如下:

```
void sort (array array[,int sort_flags])
```

参数 array 表示要排序的数组,可选的第二个参数 sort_flags 可以用以下值改变排序的行为,其值和含义如下:

(1) 值 SORT_REGULAR 表示正常比较数组元素,即以数组元素原来的数据类型进行比较,其为默认值。

(2) 值 SORT_NUMERIC 表示数组元素被作为数字来比较。

(3) 值 SORT_STRING 表示数组元素被作为字符串来比较。

下面是数组排序的范例。

【例 3-5-1】 (代码位置:\3\testArraySort.php)

```
<?php
  $colors=array();
  $colors[10]='red';
  $colors[11]='blue';
  $colors[12]='green';
  $colors[13]='yellow';
  sort($colors);
  foreach($colors as $key=>$color){
      echo "Do you like $color? <br />";
  }
?>
```

例中以正常的方式排序数组 $colors 中的元素,程序运行结果如下:

```
Do you like blue?
Do you like green?
Do you like red?
Do you like yellow?
```

3.5.2 逆排序

使用 rsort 函数进行逆排序,下面是函数 rsort 进行逆排序的范例。

【例 3-5-2】 (代码位置:\3\testArrayRSort.php)

```
<?php
  $colors=array();
  $colors[10]='red';
  $colors[11]='blue';
```

```
    $colors[12]='green';
    $colors[13]='yellow';
    rsort($colors);
    foreach($colors as $key=>$color){
      echo "Do you like $color? <br />";
    }
?>
```

程序运行结果如下：

```
Do you like yellow?
Do you like red?
Do you like green?
Do you like blue?
```

3.5.3 打乱数组排序

函数 shuffle 用来打乱一个数组(即随机排列数组元素的顺序)。其格式如下：

```
void shuffle (array array)
```

下面是使用函数 shuffle 打乱数组排序的范例。

【例 3-5-3】 (代码位置：\3\testArrayShuffle.php)

```
<?php
    $colors=array();
    $colors[10]='red';
    $colors[11]='blue';
    $colors[12]='green';
    $colors[13]='yellow';
    shuffle($colors);
    foreach($colors as $key=>$color){
      echo "Do you like $color? <br />";
    }
?>
```

上例每次运行数组 $ colors 的元素次序将发生变化,其输出结果都会不同。

3.5.4 删除数组元素

函数 array_splice 的作用是把数组中的一部分去掉并用其他值取代。其格式如下：

```
array array_splice ( array input,int offset [,int length [,array replacement]])
```

参数 input 表示要删除的数组,参数 offset 和 length 用来指定要删除元素的起始位置和个数。参数 input 和 offset 是必选的,length 是可选的。如果提供了 replacement 参

数，则用数组 replacement 中的元素取代数组 input 中被删除的元素。该函数会返回一个包含有被移除元素的数组。如果 offset 为正，则从数组 input 中该值指定的偏移量开始移除。如果 offset 为负，则从数组 input 末尾倒数该值指定的偏移量开始移除。如果省略参数 length，则移除数组中从 offset 到结尾的所有部分。如果 length 为正值，则移除 length 个元素。如果 length 为负值，则移除从 offset 到数组末尾倒数 length 为止中间所有的元素。

下面是删除指定位置的一个数组元素的范例。

【例 3-5-4】 （代码位置：\3\testArraySplice1. php）

```
<?php
  $colors=array();
  $colors[10]='red';
  $colors[11]='blue';
  $colors[12]='green';
  $colors[13]='yellow';
  array_splice($colors,2,1);          //删除下标为 2 的元素
  foreach($colors as $key=>$color){
    echo "Do you like $color? <br />";
  }
?>
```

例中指定要删除数组 $ colors 中的元素，从下标为 2 的元素开始删除，删除元素的个数是 1。程序运行结果如下：

```
Do you like red?
Do you like blue?
Do you like yellow?
```

下面的范例从数组指定位置删除之后的所有数组元素。

【例 3-5-5】 （代码位置：\3\testArraySplice2. php）

```
<?php
  $colors=array();
  $colors[10]='red';
  $colors[11]='blue';
  $colors[12]='green';
  $colors[13]='yellow';
  array_splice($colors,2);            //从下标 2 开始删除之后的元素
  foreach($colors as $key=>$color){
      echo "Do you like $color? <br />";
  }
?>
```

例中将从数组 $ colors 中的第二个元素开始删除，由于未指定删除元素个数，因此默认删除其后的所有元素，程序将删除元素 $ colors[12] 和 $ colors[13]。程序运行结果

如下：

```
Do you like red?
Do you like blue?
```

下面的范例反向删除数组元素。

【例 3-5-6】 （代码位置：\3\testAraySplice3. php）

```
<?php
  $colors=array();
  $colors[10]='red';
  $colors[11]='blue';
  $colors[12]='green';
  $colors[13]='yellow';
  array_splice($colors,-2);            //从数组最后一个元素反向删除 2 个元素
  foreach($colors as $key=>$color){
    echo "Do you like $color? <br />";
  }
?>
```

例中从数组＄colors 倒数第二个元素开始删除元素，由于未指定删除元素的个数，因此默认将删除其后面所有的元素，程序将删除元素＄colors[12]和＄colors[13]。程序运行结果如下：

```
Do you like red?
Do you like blue?
```

第 4 章



PHP 函数

编程过程通常会借用函数实现某种功能。一个函数的作用就相当于一台机器,不同的机器其作用会各不相同,就像放玉米进机器中,它加工出来的是爆米花。不同的函数能完成不同的特定的功能。PHP 函数(function)是指一段完成指定任务的已命名代码,函数可以遵照给它的一组值或参数完成任务,并且可能返回一个值。使用函数可以节省编译时间,因为无论调用多少次,函数只需被编译一次。当程序代码多了以后,如何组织这些程序?PHP 最初的设计原则是用函数来组织,这样可以让一段代码形成一个“程序模块”,不管在什么地方使用到相同功能时都可调用该函数,省去了重复编写代码的麻烦。函数可以把不同功能的代码独立到一处,减少耦合性,提高代码可重用性和增强程序的安全性。此外,函数还可以把代码和其他已实现的函数整合起来实现功能更加强大的函数。PHP 函数可分用户自定义函数和系统内置函数。PHP 中一个函数由 4 部分组成:函数名、参数、函数体和返回值。

4.1 用户自定义函数的定义

PHP 在提供大量内置函数的基础上,还开放了自定义函数的功能,允许开发者按照功能需求来编写实现特定目的的自定义函数。用户自定义函数的语法如下:

```
function functionname($args)
{
    表达式
}
```

function 为函数声明时的关键字;functionname 是要声明的函数的名称;参数 $ args 表示外部传递给函数的参数变量,参数 $ args 是可选的,而且可以是多个参数;表达式为函数要实现的功能主体,任何符合 PHP 语法的语句都可以作为表达式。

函数名是函数在程序代码中的识别名称,函数名可以是以字母或下划线开头后跟零个或多个字母、下划线和数字的任何字符串。函数名不区分大小写。给函数命名时不可使用已声明的函数名称或 PHP 内建的函数名称。函数外部可以通过参数 $ args 把数据传递给函数体中的表达式加以运算处理,多个参数之间要用逗号隔开。当函数不需要任

何数值传入时,可以省略参数。下面是自定义函数的范例。

```
<?PHP
  function myfun($name)
  {
    echo $name;
    echo "hello";
    return "$name : hello";
  }
?>
```

上例中,function是定义函数的关键字,myfun是函数名,$name是函数参数,花括号中的代码是函数体,return表示函数有返回值。

4.2 函数的调用

用户定义好函数后,就可以在程序中直接调用该函数了,函数的调用方法很简单,格式如下:

```
functionname($args)
```

functionname为要调用的函数名称,该函数需要已定义好,参数$args是传递给函数的值。调用函数时,直接用一个函数的函数名加一对半角括号即可。如果函数需要参数,则要在括号中给出函数所需要的参数表。在PHP中,用户自定义函数可以写在被调用位置的前面,也可以写在被调用位置的后面。这种调用方式也适用于调用系统内置函数。下面是调用函数的实例,在该例中,myfun和sayHello是自定义函数,echo是系统定义的函数。

【例4-2-1】 (代码位置:\4\testCallFun.php)

```
<?PHP
  function myfun()
  {
      echo "hello";
  }
  myfun();                //调用自定义的函数 myfun
  sayHello();             //调用自定义的函数 sayHello
  function sayHello()
  {
      echo "Hello";
  }
?>
```

例中函数myfun的定义写在被调用位置的前面,而函数sayHello的定义写在被调用位置的后面。程序运行后浏览器输出:helloHello。

4.3 函数参数

有时主程序需要动态传递不同的值给函数处理，通过参数列表可以传递信息到函数。PHP中传递参数的方式有按值传递参数和按址传递参数。按值传递参数是指父程序直接传递指定的值或变量给函数使用。由于所传递的值或变量与函数里的数值分别储存于不同的内存区块，所以当函数对所导入的数值作了任何变动时，并不会对父程序造成直接影响。这就好比用户A把资料的复印件给用户B使用，资料的原件和复印件互相独立，复印件(或原件)的损坏并不影响原件(或复印件)，两者互不影响。相对于按值传递模式，按址传递参数并不会将父程序中的指定数值或目标变量传递给函数，而是把该数值或变量的内存储存区块相对地址导入函数之中。因此当该数值在函数中有任何变动时，会连带对父程序造成影响。下面是按值传递参数的范例。

【例 4-3-1】 (代码位置：\4\testFunVParam.php)

```
<?PHP
  sayHello("小王");
  function sayHello($name)
  {
       echo "$name,Hello";
  }
?>
```

程序运行结果如下：

```
小王,Hello
```

下例是按值传递数组参数的范例。

【例 4-3-2】 (代码位置：\4\testFunArrayParam.php)

```
<?php
  function takes_array($input)
  {
    echo "$input[0] +$input[1]=", $input[0]+$input[1];
  }
  $myvals=array(10,20);
  takes_array($myvals);
?>
```

程序运行结果如下：

```
10+20=30
```

下面是按址传递参数的范例，函数print_A有两个参数变量$A和$B，前者是按值传递，后者是按地址传递，函数print_A的功能是改变两者的值并打印其值，由于参数$B是按地址传递，而参数$A是按值传递，所以主程序中的变量$B也被改变，而$A

没有被改变。

【例 4-3-3】 （代码位置：\4\testFunRefParam.php）

```
<?php
    $A="Today";
    $B="Monday";
    function print_A($A,&$B)
    {
        $A="今天";                          //修改变量$A的值
        $B=$A." is ".$B;                    //修改变量$B的值
        echo "函数中变量 A 与变量 B 的值为<br>";
        echo "变量 A: $A <br>";
        echo "变量 B: $B <p>";
    }
                                            //以传址方式导入变量 B
    print_A($A,$B);
    echo "主程序中变量 A 与变量 B 的值为<br>";
    echo "变量 A: $A <br>";
    echo "变量 B: $B <p>";
?>
```

程序运行结果如下所示，从结果中可知，变量 $A 没有被改变，而变量 $B 被改变了。

```
函数中变量 A 与变量 B 的值为
变量 A: 今天
变量 B: 今天 is Monday

主程序中变量 A 与变量 B 的值为
变量 A: Today
变量 B: 今天 is Monday
```

如果在调用函数时没有指定参数的值，在函数中会使用参数的默认值。默认参数必须列在所有没有默认值参数的后面。下面是默认值传递参数的范例，函数 fun_sum 中的变量 $b 和 $c 默认值是 0，它们必须写在参数 $a 的后面，程序运行输出 30 和 60。

【例 4-3-4】 （代码位置：\4\testSTParam.php）

```
<?php
  function fun_sum($a,$b=0,$c=0)
  {
    return $a+$b+$c;
  }
  echo fun_sum(10,20);          //$b的值是 20,而变量$c的是 0
  echo "<br/>";
  echo fun_sum(10,20,30);       //变量$b和$c值分别是 20和 30,而不是 0
?>
```

4.4 函数返回值

外部程序通过参数变量传递数据给函数,函数处理数据后会返回结果给外部程序使用,外部程序和函数体之间就是通过参数变量和返回值来交互的。函数运行的结果是通过返回值反馈给父程序的。如果在一个函数中调用 return 语句,则立即结束此函数的执行并将 return 的参数作为函数的值返回。函数返回值用法如下:

```
function functionname(变量)
{
    表达式;
    return 变量;
}
```

下面是关于函数返回值的范例。

【例 4-4-1】 (代码位置:\4\testFunRet.php)

```
<?php
    echo "用函数求累加:<br>";
    function test($a)
    {
        $sum=0;
        for($i=0;$i<=$a;$i++)
            $sum+=$i;
        return $sum;            //也可写成 return($sum)
    }
    echo "5的累加是:".test(5)."<br>";
?>
```

上例函数 test 会累加 0 至 $a 的和,并把累加和作为返回值返回给主程序使用。程序运行结果如下:

```
用函数求累加:
5的累加是:15
```

注意:return 是语言结构而不是函数,并不需要用括号将参数括起来。事实上不用括号比用括号更常见,其实用哪一种并无所谓。如果函数中没有 return 则函数结束时不会有返回值。函数也可以带有 return 而不返回任何值,下面是无返回值的范例。

【例 4-4-2】 (代码位置:\4\testFunVoidRet.php)

```
<?php
    function test()
    {
        echo "hello";
        return ;                //无返回值
```

```
    }
    test();
?>
```

关于 return 还有其他值得注意的地方,它并不局限于只用在函数中,还可以使用在其他地方。如果在全局范围中调用,则当前脚本文件中止运行。如果当前脚本文件是被 include 的或者 require 的,则控制交回调用文件。此外,如果当前脚本是被 include 的,则 return 的值会被当作 include 调用的返回值。如果在主脚本文件中调用 return,则脚本中止运行。

4.5 内 置 函 数

PHP 内置函数是指 PHP 预提供的函数,不需要用户定义即可使用。内置函数可分为标准函数和扩展函数。标准函数是指编译到 PHP 内核中的函数,可以直接在程序中使用,不需要其他任何声明和加载库工作。扩展函数按照功能的不同被分门别类地封装在多个函数库中,扩展函数一般并不能在代码中直接使用,而是要按照程序需求有选择地开启和关闭。打开一个扩展函数库的方法很简单,只需打开 PHP 安装目录下的文件 php-apache2handler.ini,将文件中“;extension=php_xxx.dll”前面的分号“;”去掉,并且保存 php-apache2handler.ini 和重新启动 Apache Web 服务器即可。常用的数据库和 SOAP 等扩展函数库都可通过这种方法启动和关闭。本节主要介绍标准函数。

4.5.1 die 函数

函数 die 用来输出一条消息,并退出当前脚本。die 的语法如下:

```
die(status)
```

如果参数 status 是字符串,则该函数会在退出前输出字符串。如果 status 是整数,这个值会被用作退出状态。退出状态的值为 0～254。退出状态 255 由 PHP 保留,不会被使用。状态 0 用于表示成功地终止程序。exit 和 die 相同。下面是函数 die 的范例。

【例 4-5-1】 (代码位置:\4\testDie.php)

```
<?php
  $a=false;
  $a or die("false");             //变量$a 为 false 则执行 die 语句输出 false
  echo "true";                    //该语句不会被执行
?>
```

4.5.2 日期时间函数

函数 date 用来格式化一个本地时间/日期,其格式如下:

```
date(format [,timestamp])
```

整型参数 timestamp 表示时间截,是可选的,字符串 format 规定如何返回结果,例如,h 表示小时,i 表示分钟,s 表示秒。下面是函数 date 的范例。

【例 4-5-2】 (代码位置:\4\testDate. php)

```
< ?php
  //Assuming today is: March 10th, 2001, 5:16:18 pm
  $today= date("F j, Y, g:i a");
  echo "$today <br/> ";
  $today= date("m.d.y");
  echo "$today <br/> ";
  $today= date("j, n, Y");
  echo "$today <br/> ";
  $today= date("Ymd");
  echo "$today <br/> ";
  $today= date('h- i- s, j- m- y, it is w Day z ');
  echo "$today <br/> ";
  $today= date('\i\t \i\s \t\h\e jS \d\a\y.');
  echo "$today <br/> ";
  $today= date("D M j G:i:s T Y");
  echo "$today <br/> ";
  $today= date('H:m:s \m \i\s\ \m\o\n\t\h');
  echo "$today <br/> ";
  $today= date("H:i:s");
  echo "$today <br/> ";
?>
```

程序运行结果如下:

```
August 9, 2012, 8:09 am
08.09.12
9, 8, 2012
20120809
08- 09- 32, 9- 08- 12, 0931 0932 4 Thuam12 221
it is the 9th day.
Thu Aug 9 8:09:32 UTC 2012
08:08:32 m is month
08:09:32
```

函数 microtime 的主要作用是返回当前 UNIX 时间戳和微秒数。不过这个函数仅在支持 gettimeofday()系统调用的操作系统下可用。下面是关于函数 microtime 的范例,该范例先随机产生循环次数,然后获取开始循环时间,再执行循环,接着获取循环结束时间,最后通过计算两个时间的时间差获取循环的运行时间。

【例 4-5-3】（代码位置：\4\testMicrotime.php）

```
<?php
    //函数 fn 功能：获取时间
    function fn(){
      list($a,$b)=explode(' ',microtime());      //获取当前时间戳和微秒数
      return $a+$b;                              //相加并返回
    }
    $times=rand(10000, 1000000);                 //随机生成运行次数
    $start_time=fn();                            //获取开始时间
    for($i=1;$i<$times;$i++){          //用 for 循环执行$times 次,这个执行会耗用一定时间
    }
    $over_time=fn();                             //获取结束时间
    //$over_time-$start_time 是时间差,round 函数对时间差保留 3 位小数
    echo '您本次页面执行时间为：'.round($over_time-$start_time,3).'秒';
?>
```

上例由于循环的次数是随机产生的，因此每次执行时间会有所不同。本书执行的结果如下：

```
您本次页面执行时间为：0.035 秒
```

4.5.3 sleep 函数

函数 sleep 用来使程序延迟执行指定的秒数。格式如下：

```
int sleep (int $seconds)
```

函数成功执行时返回 0，错误时返回 FALSE，参数 $ seconds 表示时间，以秒计时。下面是关于 sleep 的范例。

【例 4-5-4】（代码位置：\4\testSleep.php）

```
<?php
  //显示当前时间
  echo date('h:i:s')."\n";
  echo "<br/>";
  //等待 10 秒
  sleep(10);
  //显示当前时间
  echo date('h:i:s')."\n";
?>
```

程序运行结果如下：

```
12:22:25
12:22:35
```

4.5.4 字符串函数

PHP 字符串函数是必不可少的,这给解决实际问题提供了很多便利,比如使用 PHP 字符串函数对字符串分割、截取、匹配和替换等处理。下面是关于字符串函数 trim 的范例,trim 用来去除字符串左右两边的空格。

【例 4-5-5】 (代码位置:\4\testTrim.php)

```
<?php
  $str=" He llo     ";                          //长度为 11
  echo strlen($str);                            //显示去除空格前的长度
  echo "<br/>";
  $str=trim($str);                              //去除字符串前后的空格
  echo strlen($str);
?>
```

程序运行后,会去除字符串“ He llo   ”中的前后空格,去除前后的字符串长度值分别为 11 和 6。下面是多个字符串函数的范例。

【例 4-5-6】 (代码位置:\4\testStrOPS.php)

```
<?php
  $pizza="第一个元素 第二个元素 第三个元素 第四个元素";
  $pieces=explode(" ", $pizza);                 //字符串返回到数组变量中
  print_r($pieces);
  echo "</br>";
  $array=Array("ABC","123","测试");
  $colon_separated=implode(":", $array);        //将数组的内容组合成一个字符串
  echo($colon_separated);
  echo "</br>";
  $str=strrev("ABCDEFG");                       //将字符串前后颠倒
  echo $str;
  echo "</br>";
                                                //将%body%以 black 取代
  $bodytag=str_replace("%body%", "black", "This is a body. text=%body%");
  echo $bodytag;
?>
```

程序运行结果如下:

```
Array([0]=>第一个元素 [1]=>第二个元素 [2]=>第三个元素 [3]=>第四个元素 )
ABC:123:测试
GFEDCBA
This is a body. text=black
```

表 4-5-1 列出了常用的字符串函数。

表 4-5-1 字符串函数

函 数 名	作 用
substr_replace	把字符串的一部分替换为另一个字符串
substr_count	计算子串在字符串中出现的次数
substr_compare	从指定的开始长度比较两个字符串
substr	返回字符串的一部分
strtr	转换字符串中特定的字符
strtoupper	把字符串转换为大写
strtolower	把字符串转换为小写
strtok	把字符串分割为更小的字符串
strstr	搜索一个字符串在另一个字符串中的第一次出现
strspn	返回在字符串中包含的特定字符的数目
strrpos	查找字符串在另一个字符串中最后一次出现的位置,其对字符串区分大小写
strripos	查找字符串在另一个字符串中最后一次出现的位置,其对字符串不区分大小写
strrev	反转字符串
strrchr	查找字符串在另一个字符串中最后一次出现的位置,并返回从该位置到字符串结尾的所有字符
strpos	返回字符串在另一个字符串中第一次出现的位置
strpbrk	在字符串中搜索指定字符中的任意一个
strncmp	比较两个字符串,字符串要区分大小写
strncasecmp	比较两个字符串,字符串不区分大小写
strnatcmp	使用一种"自然"算法来比较两个字符串

4.5.5 文件操作函数

下面是 PHP 删除文件的范例,先创建一个文件,其名为 test. txt,然后再删除该文件。

【例 4-5-7】 (代码位置:\4\testUnlink. php)

```
<?php
  $myFile="test.txt";
  $fh=fopen($myFile, 'w')or die("can not open file");
  fclose($fh);
  //删除文件
  unlink($myFile);
?>
```

运行程序,如果程序正常运行,那么系统会先自动创建一个 0 字节的文件 test. txt,然后再删除该文件。如果文件 test. txt 已存在,那么系统先将文件长度置零,然后再删除文件。如果文件不能被创建,如文件为只读,那么将提示错误。运行结果如下:

```
Warning: fopen(test.txt)[function.fopen]: failed to open stream: Permission denied in C:\
PHPnow-1.5.6\htdocs\testUnlink.php on line 3
can not open file
```

下面是删除文件夹的范例。例中自定义了一个函数 deldir，该函数有一个参数 $ dir，该参数表示要删除的目录名，函数的作用是删除指定文件夹中的子文件夹及文件。

【例 4-5-8】 （代码位置：\4\testDeldir.php）

```
<?php
  function deldir($dir){
    $dh=opendir($dir);
    while($file=readdir($dh)){
      if($file!="." && $file!=".."){
        $fullpath=$dir."/".$file;
        if(!is_dir($fullpath)){
          unlink($fullpath);
        } else {
          deldir($fullpath);
        }
      }
    }
    closedir($dh);
    if(rmdir($dir)){
      return true;
    } else {
      return false;
    }
  }
  deldir("新建文件夹");
?>
```

程序运行后将会自动删除当前文件夹下的“新建文件夹”。

下面是获取文件信息的范例。例中通过已打开的文件句柄 $ fp 取得文件的统计信息，文件统计信息由函数 fstat 实现。

【例 4-5-9】 （代码位置：\4\testFstat.php）

```
<?php
  //打开文件
  $fp=fopen("test.txt", "r");
  //取得统计信息
  $fstat=fstat($fp);
  //关闭文件
  fclose($fp);
  //只显示关联数组部分
  print_r(array_slice($fstat,13));
```

```
?>
```

程序运行后将显示文件 test. txt（文件须存在）的统计信息，运行结果如下：

```
Array([dev]=>0 [ino]=>0 [mode]=>33206 [nlink]=>1 [uid]=>0 [gid]=>0 [rdev]=>0 [size]=>3
[atime]=>1387081386 [mtime]=>1387081397 [ctime]=>1387081386 [blksize]=>-1 [blocks]=>-
1)
```

4.5.6 数学函数

在实际业务中，对数据进行数学运算是必不可少的，下面是数学函数的范例。

【例 4-5-10】（代码位置：\4\testMath. php）

```
<?php
  echo pow(2, 4);                    //计算 4 的平方
  echo " ";
  echo round(19.12345,3);            //保留 3 位小数
  echo " ";
  echo rand(10, 1000);               //随机产生 10~1000 的数
  echo " ";
  echo max(1,2,100,9,30);            //取最大值
  echo " ";
  echo min(4,10,-9,20);              //取最小值
?>
```

程序运行结果如下：

```
16 19.123 855 100 -9
```

常见数学函数如表 4-5-2 所示。

表 4-5-2 常用数学函数

函数名	作　　用	函数名	作　　用
tanh	返回一个数的双曲正切	hexdec	把十六进制转换为十进制
tan	返回一个数的正切	exp	计算 e 的指数（e 为 2.718282）
sqrt	返回一个数的平方根	deg2rad	将角度转换为弧度
sinh	返回一个数的双曲正弦	decoct	把十进制转换为八进制
sin	返回一个数的正弦	dechex	把十进制转换为十六进制
round	对浮点数进行四舍五入	decbin	把十进制转换为二进制
rand	返回随机整数	cosh	返回一个数的双曲余弦
rad2deg	把弧度数转换为角度数	cos	返回一个数的余弦
pi	返回圆周率的值	ceil	向上舍入为最接近的整数
log10	以 10 为底的对数值	abs	返回一个数的绝对值

第 5 章



面向对象开发

生活中的系统可以看成由多个实体组成，例如，多媒体教学系统由一间教室、多张椅子、多台计算机和路由器等实体组成。一台电视机是实体，电视机是根据电视机设计图生产出来的，换句话说，实体是根据设计图被生产出来的。面向对象编程是目前流行的一种编程思想，它通过使用对象模拟实体和使用类模拟设计图来实现系统的功能。对象(object)是类(class)的一个实例(instance)。如果将对象比作房子，那么类就是房子的设计图纸。所以面向对象程序设计的重点是类的设计，而不是对象的设计。

5.1 类的定义

每个类的定义都以关键字 class 开头，后面跟着类名，类名可以用任何非 PHP 保留字，为了方便阅读，一般类名首字符要大写。类名后面跟着一对花括号，里面包含类的成员和方法的定义。下面定义了一个类，类名是 Page。

```
<?php
  class Page
  {
  }
?>
```

用来描述对象的数据元素称为对象的属性(也称为数据/状态)，对对象属性的操作称为对象的方法(也称为行为/操作)。在 PHP 5 中，属性指在 class 中声明的变量，方法是指 class 中声明的函数。PHP 类属性和方法定义如下：

```
<?php
  class Page
  {
      //定义两个成员变量$name 和$color
      var $name;
      var $color;
      //定义两个成员变量 echoName 和 echoColor
      function echoName()
```

```
        {
        }
        function echoColor()
        {
        }
    }
?>>
```

上例在类 Page 中定义了 $name 和 $color 两个成员属性，以及 echoName 和 echoColor 两个成员方法。

5.2 创建和使用对象

对象是通过类的实例化创建出来的，当定义好类后，使用 new 关键字来生成一个对象。下面是创建对象的范例，例中创建了一个 Page 类型的对象，并调用了对象的方法 echoName。

【例 5-2-1】 （代码位置：\5\testCreateObj. php）

```
<?php
  class Page
  {
      //定义两个成员变量$name 和$color
      var $name;
      var $color;
      //定义两个成员变量
      function echoName()
      {
      }
      function echoColor()
      {
      }
  }
  $o1=new Page();                    //创建一个 Page 类型对象
  $o1->echoName();                   //访问对象的方法
  $o2=new Page();                    //创建另一个 Page 类型对象
  echo $o2->color;                   //访问对象的属性
?>
```

从上例可知，一个类可以实例化多个对象，每个对象都是独立的。对象的变量和方法可使用“对象->属性”、“对象->方法”的形式访问。

5.3 $this 关键字

一个对象访问另一个对象是通过“对象->成员”形式来实现的。在对象的内部，对象的方法要访问本对象的属性，或是对象的方法调用本对象的其他方法，其还需要通过“对

象->成员"形式访问吗？回答是否定的。PHP提供了对象访问自己的访问形式。因为对象的所有成员都要用对象来调用，包括对象的内部成员之间的调用，所以PHP提供了一个本对象的引用$this，每个对象都用本对象的引用$this来代表这个对象，用其来完成对象内部成员的调用，$this的本意就是"自身"的意思。下面是对象自己访问自己的范例。

【例 5-3-1】 （代码位置：\5\testThis.php）

```
<?php
  class Page
  {
      //定义两个成员变量$name和$color
      var $name;
      var $color;
      //定义两个成员变量
      function echoName()
      {
        echo $this->name;           //访问对象自身的属性
      }
      function echoColor()
      {
        echo $this->color;          //访问对象自身的属性
      }
  }
  $o1=new Page();
  $o1->echoName();
  $o2=new Page();
  echo $o2->color;
?>
```

例中创建了o1和o2两个对象，这两个对象的方法echoName和echoColor分别通过$this访问了对象自己的属性$name和$color。从本例可知，对象可以通过"$this->变量名"和"$this->方法名"来访问对象自身的属性和方法。

5.4 构造函数

到目前为止，上述例子还没有给对象的属性赋初值，PHP可以通过构造函数给对象的属性赋初值。构造函数又称为构造方法，是对象被创建时自动调用的方法，用来完成类初始化的工作。大多数类都有一种称为构造函数的特殊方法。当创建一个对象时，它将自动调用构造函数，也就是使用new这个关键字来实例化对象的时候自动调用构造方法。构造函数的声明与其他操作的声明一样，只是其名称必须是__construct（__是两个下划线）。这是PHP 5中的变化，以前的版本中，构造函数的名称必须与类名相同，这在PHP 5中仍然可以用，但现在已经很少有人用了，这样做的好处是可以使构造函数独立于类名，当类名发生改变时不需要改变相应的构造函数名称。构造函数可以不带参数也

可以带参数。值得注意的是：在 C++ 等开发语言中，构造函数是可以重载定义多个的，但在 PHP 中，在一个类中只能声明一个构造方法。下面是带有参数构造函数的范例。

【例 5-4-1】（代码位置：\5\testConstruct.php）

```
<?php
    class Page
    {
        //定义两个成员变量$name 和$color
        var $name;
        var $color;
        //定义两个成员变量
        function echoName()
        {
          echo "$this->name <br/>";
        }
        function echoColor()
        {
          echo "$this->color <br/>";
        }
        function __construct($name,$color)
        {
          $this->name=$name;
          $this->color=$color;
        }
    }
    $o1=new Page("page1","red");
    $o1->echoName();
    $o1->echoColor();
    $o2=new Page("page2","blue");
    echo $o2->name;
    echo $o2->color;
?>
```

上例创建了两个对象，在创建对象时自动调用构造函数，并给属性赋初值：page1 和 red，page2 和 blue。程序运行结果如下：

```
page1
red
page2blue
```

5.5　析构函数

在销毁一个类对象之前需要执行的一些操作或完成一些功能，比如说关闭文件或释放数据库结果集等，在 PHP 中可以通过析构函数来实现。析构函数会在到某个对象的

所有引用都被删除或者当对象被显式销毁时执行，也就是对象在内存中被销毁前调用析构函数。析构函数不能带参数。析构函数的名称是__destruct(__是两个下划线)。在脚本执行结束时，PHP 会销毁内存中的对象，但有些对象外部的资源比如 Cookie 等，就应当用此函数销毁。下面是析构函数范例。

【例 5-5-1】（代码位置：\5\testDestruct. php）

```
<?php
    class Page
    {
        //定义两个成员变量$name 和$color
        var $name;
        var $color;
        //定义两个成员变量
        function echoName()
        {
          echo $this->name;
        }
        function echoColor()
        {
          echo $this->color;
        }
        function __construct($name,$color)
        {
          $this->name=$name;
          $this->color=$color;
        }
        function __destruct()
        {
          echo "自动执行析构函数<br/>";
        }
    }
    $o1=new Page("page1","red");
    $o2=new Page("page2","blue");
?>
```

上例创建了两个对象，当脚本执行结束后，系统要删除对象，这里有两个对象要被删除，在删除对象前会自动执行析构函数，程序运行结果如下：

```
自动执行析构函数
自动执行析构函数
```

5.6 访问控制

有些实体属性允许被外部访问，有些则不允许，比如，影视明星实体的获奖等属性允许被公众访问，年龄等属性不允许被公众访问，但可以被亲朋好友访问。从类的角

度看就是,有些类对象的变量不允许被外部访问,有些类对象的变量允许其他对象访问。访问控制是指允许/不允许访问类对象的变量和方法的方式。PHP 为类变量提供了 3 种访问控制方式:公有、私有和受保护方式,分别用 public、private 和 protected 来修饰。

5.6.1 private 访问控制

被 private 修饰的变量不能直接被对象外部调用,只能由当前对象调用,下面是关于 private 的范例。

【例 5-6-1】 (代码位置:\5\testPrivate. php)

```
<?php
  class Money
  {
      private $mymoney=1000;
      function echoMymoney()
      {
        echo $this->mymoney;
      }
  }
  $mon=new Money();
  $mon->echoMymoney();
  echo $mon->mymoney;          //私有变量被对象外部访问,抛出异常
?>
```

上例运行后会显示 1000,同时,还会出现异常提示:Cannot access private property Money。从例中可知,类变量 $ mymoney 可以被类方法 echoMymoney 访问,即私有成员能被自身访问,但不允许类外部访问。

5.6.2 public 访问控制

被 public 修饰的属性和方法可以被类和类外部无限制地访问,类变量和方法的默认访问权限是 public 访问权限。下面是 public 访问权限范例。

【例 5-6-2】 (代码位置:\5\testPublic. php)

```
<?php
  class Money
  {
    public $mymoney=1000;
    function echoMymoney()
    {
      echo $this->mymoney;
    }
  }
```

```
    $mon=new Money();
    $mon->echoMymoney();
    echo $mon->mymoney;
?>
```

程序运行后会显示出类变量$mymoney的值，不会出现异常。从例中可知，public修饰的变量可以被类自身和类外部访问。

5.6.3 protected访问控制

从类外部访问角度看，它和private是相同的效果，经它修饰的变量不允许类外部访问。但从继承角度看，protected和private效果却不同，具体参看5.7节的内容。

5.7 类的继承

5.7.1 类继承

彩色电视机继承了黑白电视机的功能，同时还拥有了黑白电视机所不具备的彩色显示功能。一个师傅带多个徒弟，不同的徒弟可以学(继承)师傅的部分或全部手艺，同时，徒弟还可以创新或拥有师傅未掌握的手艺。PHP提供类继承解决和“徒弟传承师傅手艺”相似的这种问题。继承是类的一个强大功能，通过继承，一个类可以不需要编写代码就能拥有另一个类的功能，这样就可以避免编写相同的代码，具有工作效率高和可维护性好等优点。如果一个类A从另一类B中继承而来，那么类A就拥有了类B的功能，称类A为子类或派生类，称类B为父类或基类。派生类将包含有基类的所有属性和方法，并可以在派生类中加上其他属于自己的属性和方法，也可以覆写基类的方法和属性。可以用extends关键字来继承一个类。下面是类继承范例。

【例5-7-1】 (代码位置：\5\testExtends.php)

```
<?php
    class Person{                   //定义父类
        function eat()
        {
            echo "吃饭";
        }
    }
    class Stu extends Person{       //继承父类
    }
    $o=new Stu();
    $o->eat();
?>
```

上例定义了父类Person，该类中有一个方法eat，由于Stu也是人，也同样具有eat行

为，因此可以不用重写 eat 方法而直接继承 Person，从而具有了 eat 方法。运行结果如下：

```
吃饭
```

5.7.2 构造函数的继承

当父类有构造函数而子类没有，则子类会在实例化时自动执行父类的构造函数，此时，子类需要引入父类构造函数中所需的参数，否则出错。PHP 创建子类的对象时，当子类没有构造函数，则会从实例所在的子类向上（即父类）搜索合适的构造函数，一旦找到就停止，即使用该构造函数而不会再向上搜索，因此，子类本身如果没有构造函数，则以其最靠近的一个超类并且有构造函数的为准。当子类也有构造函数时，不管父类是否有构造函数，都会执行子类自己的构造函数。下面是构造函数继承的范例。

【例 5-7-2】（代码位置：\5\testConstructExtends.php）

```
<?php
  class Person{                          //人类
    var $color;
    function __construct($color="Red")
    {
        $this->color=$color;
    }
  }
  class Stu extends Person{              //学生类
    function __construct($color="Yellow")
    {
        $this->color=$color;
    }
  }
  class Pupil extends Stu
  {//小学生
    function eachoColor()
    {
        echo "$this->color";
    }
  }
  $o=new Pupil();
  $o->eachoColor();
?>
```

上例有 Person、Stu 和 Pupil 三个类，Person 是 Stu 的父类，Stu 是 Pupil 的父类，程序创建类 Pupil 的对象时，由于该类没有构造函数，程序将向上寻找构造函数，由于 Stu 是拥有构造函数且离 Pupil 最近的父类，因此自动调用 Stu 的构造函数，而不会再继续向

上搜索执行Person的构造函数。运行结果如下：

```
Yellow
```

5.7.3 在子类中调用父类的属性或方法

类通过$this访问类自己的变量和方法，子类如何访问父类的变量和方法呢？可以通过“parent::父类方法名或变量名”或“父类名::父类方法或变量名”的形式来访问，parent是子类访问父类属性和方法的关键字。下面是子类调用父类方法的范例。

【例5-7-3】 （代码位置：\5\testParent.php）

```
<?php
  class Person{                          //人类
    var $color;
    function __construct()
    {
        echo "Person的构造函数";
    }
  }
  class Stu extends Person{              //学生类
    function __construct($color="Yellow")
    {
        parent::__construct();
        Person::__construct();
    }
  }
  $o=new Stu();
?>
```

例中，类Stu的构造函数分别使用parent::__construct()和Person::__construct()两种形式调用父类的方法，运行结果如下：

```
Person的构造函数
Person的构造函数
```

5.7.4 访问控制的继承

父类的变量受public、protected和private修饰后，经子类继承后变量的访问控制情况会不同。下面是public的继承范例。

【例5-7-4】 （代码位置：\5\testPublicExtends.php）

```
<?php
  class Person{                          //人类
    public $color;
    function __construct()
```

```
    {
        $this->color="Red";
    }
  }
  class Stu extends Person{            //学生类
    function echoColor()
    {
        echo "$this->color<br/>";
    }
  }
  $o=new Stu();
  $o->echoColor();
  echo $o->color;
?>
```

被 public 修饰的变量是可以被子类继承的，而且被继承后的变量还是 public 访问权限。变量＄color 是 public 权限，故被继承后仍是 public 权限，它可被对象外部访问，运行结果如下：

```
Red
Red
```

被 protected 修饰的变量经子类继承后，被继承的变量不允许被外部类访问但允许子类访问。下面是 protected 的继承范例。

【例 5-7-5】（代码位置：\5\testProtectedExtends. php）

```
<?php
  class Person{                        //人类
    protected $color;
    function __construct()
    {
        $this->color="Red";
    }
  }
  class Stu extends Person{            //学生类
    function echoColor()
    {
        echo $this->color;
    }
  }
  $o=new Stu();
  $o->echoColor();
  $o->color;
?>
```

上例中，变量＄color 是受保护访问的变量，因此不允许类外部访问但允许子类 Stu

访问,运行后显示 Red 并给出异常提示：Cannot access protected property Stu：：$color。

经 private 修饰的变量被继承后,类外部和子类都不允许访问。下面是 private 的继承范例。

【例 5-7-6】 （代码位置：\5\testPrivateExtends.php）

```
<?php
  class Person{                    //人类
    private $color;
    function __construct()
    {
        $this->color="Red";
    }
  }
  class Stu extends Person{        //学生类
    function echoColor()
    {
        echo $this->color;
    }
  }
  $o=new Stu();
  $o->echoColor();
  $o->color;
?>
```

例中由于访问了经 private 修饰的变量 $color,所以程序会给出错误提示 Undefined property：Stu：：$color。

5.8 类的多态

在面向对象的理论中,多态性的一般定义为：同一个操作作用于不同的类的实例,将产生不同的执行结果。即不同类的对象收到相同的消息时,将得到不同的结果。多态性允许每个对象以适合自身的方式去响应共同的消息。多态性增强了软件的灵活性和重用性。在实际的应用开发中,通常为了使项目能够在以后的时间里轻松实现扩展与升级,需要通过继承实现可复用模块进行轻松升级。在进行可复用模块设计时,就需要尽可能地减少使用流程控制语句。此时就可以采用多态实现该类设计。下面是类多态性范例。

【例 5-8-1】 （代码位置：\5\testPolymorphic.php）

```
<?php
  class employee{                          //定义员工父类
    function working(){                    //定义员工工作,需要在子类实现
        echo "本方法需要在子类中重载!";
```

```
    }
  }
  class painter extends employee{                //定义油漆工类
    public function working(){                   //实现继承的工作方法
        echo "油漆工正在刷漆!\n";
    }
  }
  class typist extends employee{                 //定义打字员类
    public function working(){
        echo "打字员正在打字!\n";
    }
  }
  class manager extends employee{                //定义经理类
    public function working(){
        echo "经理正在开会!";
    }
  }
  $o1=new painter();
  $o2=new typist();
  $o3=new manager();
  $o4=new employee();
  $o1->working();
  $o2->working();
  $o3->working();
  $o4->working();
?>
```

在上述程序中,首先定义一个 employee(员工)基类。然后定义继承自员工基类的 3 个员工类:painter(油漆工)类;typist(打字员)类;manager(经理)类。最后创建 3 个员工类对象和一个员工基类对象,并且这 4 个对象都调用同一个方法。其结果如下:

```
油漆工正在刷漆!打字员正在打字!经理正在开会!本方法需要在子类中重载!
```

从上例可发现,不同的对象对相同的消息(即调用 working 方法)进行响应,但执行的结果(working 方法的代码不同)却是不相同的。

5.9 PHP 接口

生活中,很多空调的遥控器是通用的,尽管它们分别属于不同的厂商。尽管不同厂商生产的 U 盘容量不同,但 U 盘都能被计算机识别。为什么会这样?因为它们是按标准的接口规范来实现的。在 PHP 中,接口是指定义了实现某种服务的一般规范,声明了所需的函数和常量,但不指定如何实现。之所以不给出实现,是因为不同的实体可能需要用不同的方式来实现公共的方法定义。

5.9.1 接口定义及实现

当定义接口时,接口中不定义类成员!类成员的定义完全交给实现类来完成。接口的创建使用关键字 interface 实现,通常,在接口名前面加上字母来进行标识,以便更容易辨认,接口书写格式如下:

```
interface 接口名
{
    方法的声明
}
```

下面的例子定义了一个接口 IUser,这个接口中有两个抽象方法:getName 和 setName。接口的写法和类很相似,但不同的是,接口只有函数声明而没有函数体(没有花括号),而类有函数体(有花括号)。

```
<?php
interface IUser
{
    function setName($name);
    function getName();
}
?>
```

要实现接口则要在类定义时使用关键字 implements。接口的方法全部是 public 方法。下面的例子实现了接口 IUser。

【例 5-9-1】 (代码位置:\5\testInterface.php)

```
<?php
  interface IUser
  {
      function setName($name);
      function getName();
  }
  class Stu implements IUser{
      private $name ;
      function setName($name)
      {
          $this->name=$name;
      }
      function getName()
      {
          return $this->name;
      }
  }
```

```
  $o=new Stu();
  $o->setName("小明");
  echo $o->getName();
?>
```

上例定义了 IUser 接口，接口里有两个 public 方法，类 Stu 实现了该接口，实现接口时也实现接口中定义的两个方法。程序运行结果如下：

```
小明
```

5.9.2 类与多接口

类不仅能实现一个接口，还可以同时实现多个接口。当一个类实现多个接口时，需要实现所有接口的所有方法。下例是一个类实现多个接口的范例。

【例 5-9-2】 （代码位置：\5\testClassInterface.php）

```
<?php
  interface father{
    function shuchu();
  }
  interface fam {
    function cook($name);
  }
  class test implements fam,father{              //实现两个接口
    function shuchu(){
      echo "接口继承,要实现两个抽象方法";
      echo "<br>";
    }
    function cook($name){
      echo "平时经常做饭的人是:".$name;
    }
  }
  $t=new test();
  $t->shuchu();
  $t->cook("妈妈");
?>
```

上例定义了 father 和 fam 接口，以及类 test。类 test 继承了这两个接口并实现了接口中的所有方法（shuchu 和 cook）。程序运行结果如下：

```
接口继承,要实现两个抽象方法
平时经常做饭的人是:妈妈
```

5.9.3 接口单继承

在 PHP 中，接口可以继承接口。虽然 PHP 类只能继承一个父类（单继承），但是接

口和类不同，接口可以实现多继承，可以继承一个或者多个接口。接口的继承也是使用 extends 关键字，要实现多继承，只要用逗号把继承的接口隔开即可。

需要注意的是，由于接口继承其他接口时，子接口会直接继承父接口的静态常量属性和抽象方法，所以类实现子接口时必须实现所有相关的抽象方法，包括父接口中的方法。下面是接口单继承的范例。

【例 5-9-3】（代码位置：\5\testInterfaceSExtends.php）

```
<?php
  interface father{
    function shuchu();
  }
  interface fam extends father{                     //继承 father 接口
    function cook($name);
  }
  class test implements fam{                        //实现 fam 接口
    function shuchu(){
      echo "接口继承,要实现两个抽象方法";
      echo "<br>";
    }
    function cook($name){
      echo "平时经常做饭的人是:".$name;
    }
  }
  $t=new test();
  $t->shuchu();
  $t->cook("妈妈");
?>
```

例中定义了 father 和 fam 两个接口，father 接口有 shuchu 方法，fam 接口有 cook 方法。由于 fam 继承了 father，因此 fam 接口除了自己定义的方法 cook，还继承了 father 中的方法 shuchu。当类 test 实现 fam 接口时，不仅要实现接口 fam 声明的方法 cook，而且还要实现其继承的方法 shuchu。程序运行结果如下：

```
接口继承,要实现两个抽象方法
平时经常做饭的人是:妈妈
```

5.9.4 接口多继承

一个子接口继承多个父接口，类实现子接口时除要实现子接口中定义的方法外还要实现所有父接口中的方法。下面是接口多继承的范例。

【例 5-9-4】（代码位置：\5\testInterfaceMExtends.php）

```
<?php
  interface father{
```

```
    function shuchu();
  }
  interface mother{
    function dayin($my);
  }
  interface fam extends father,mother{
    function cook($name);
  }
  class test implements fam{
    function dayin($my){
      echo "我的名字是:".$my;
      echo "<br>";
    }
    function shuchu(){
      echo "接口继承,要实现两个抽象方法";
      echo "<br>";
    }
    function cook($name){
      echo "平时经常做饭的人是:".$name;
    }
  }
  $t=new test();
  $t->shuchu();
  $t->dayin("小强");
  $t->cook("妈妈");
?>
```

例中定义有 father、mother 和 fam 三个接口,这三个接口分别定义有 shuchu、dayin 和 cook 方法。由于 fam 继承了 father 和 mother 接口,因此类 test 实现接口 fam 时,不仅要实现 fam 定义的方法,还要实现 father 和 mother 接口中的方法。程序运行结果如下:

```
接口继承,要实现两个抽象方法
我的名字是:小强
平时经常做饭的人是:妈妈
```

5.10 抽 象 类

生活中不同类型的对象可能有共同的性质,比如,人和产品是不同类型的事物,但它们有共性,那就是它们都有名称。再比如,电视机和衣服是不同类型的商品,但它们都有生产价格和产地等共同的属性。可以用类来描述共性,但问题是,共性不是对象,不应该用类来实例化这些共性。例如,考虑一个名为 Media 的类,它用来描述各种公开资料的共同性质,因为 Media 不表示真实的实体,所以不应该直接对其进行实例化。PHP 提供抽象类用来描述共性。抽象类是不能实例化的类,只能作为由其他类继承的基类。定义

抽象类需要借助关键字 abstract，格式如下：

```
abstract class 类名
{
  abstract public function_name();            //注意此处没有花括号
}
```

关键字 abstract 放在 class 前面表示类是抽象类，关键字 abstract 放在 function 前面表示方法是抽象方法，抽象方法是指只有声明但没有现实的方法。应用抽象类时需要注意几点：①抽象类通常具有抽象方法，方法没有花括号“{”和“}”；②子类继承抽象类时需要实现抽象方法；③不能从抽象类创建对象。下面给出抽象类的范例。

【例 5-10-1】 （代码位置：\5\testAbstractClass.php）

```
<?php
  abstract class Fruit {
    abstract protected function show_msg();
    //和一般的方法不同的是,这个方法没有大括号
    //不能创建这个抽象类的实例：$Shape_Rect=new Shape();
  }
  class Apple extends Fruit {
    private $name;
    private $color;

    function __construct($name, $color){
        $this->name=$name;
        $this->color=$color;
    }
    function show_msg(){
        echo $this->color .'的'. $this->name;
    }
  }
  $apple=new Apple("苹果", "红色");
  $apple->show_msg();
?>
```

上例定义了抽象类 Fruit，其中有抽象方法 show_msg。子类 Apple 继承抽象类 Fruit，实现了抽象类 Fruit 中的抽象方法 show_msg。运行结果如下：

```
红色的苹果
```

5.11 final 和 self 关键字

final 关键字可以修饰类，同样可以修饰类中的方法。如果类或者方法使用了 final，那么这就意味着类不能被继承或者子类不能覆盖父类的方法。下面是关于 final 类的

范例。

【例 5-11-1】 （代码位置：\5\testFinalClass. php）

```
<?php
  final class ParentClass
  {
      public function look()
      {
          echo "父类中的 look 方法被调用\n";
      }
      public function eat()
      {
          echo "父类中的 moreTesting 方法被调用\n";
      }
  }
      class ChildClass extends ParentClass
      {
          public function eat()
          {   echo "子类中的 eat 方法被调用\n";
          }
      }
?>
```

上例由于类 ParentClass 被关键字 final 修饰，所以类 ParentClass 不以被继承。程序运行结果如下：

```
Fatal error: Class ChildClass may not inherit from final class (ParentClass) in C:\PHPnow-1.
5.6\htdocs\5\ testFinalClass.php on line 18
```

下面是关于 final 方法的范例。

【例 5-11-2】 （代码位置：\5\testFinalClassMethod. php）

```
<?php
  class ParentClass
  {
      public function look()
      {
          echo "父类中的 look 方法被调用\n";
      }
      final public function eat()
      {
          echo "父类中的 moreTesting 方法被调用\n";
      }
  }
      class ChildClass extends ParentClass
      {
```

```
            public function eat()
            {   echo "子类中的 eat 方法被调用\n";
            }
        }
?>
```

由于方法加了 final 关键字修饰，所以方法不能被覆盖。程序运行结果如下：

```
Fatal error: Cannot override final method ParentClass::eat()in C:\PHPnow-1.5.6\htdocs\5\
testFinalClassMethod.php on line 18
```

self 是指类本身，也就是 self 是不指向任何已经实例化的对象，一般 self 用来指向类中的静态变量。如果使用类里面的静态（一般用关键字 static）成员，必须使用 self 来调用。还要注意使用 self 来调用静态变量必须使用：：（域运算符号）。下面是 self 的范例，程序运行后会输出 1。

【例 5-11-3】 （代码位置：\5\testSelf. php）

```
<?php
    class Count                    //定义一个 Count 的类
    {
        //定义一个静态变量$firstCount,并赋初值 0
        private static $firstCount=0;
        private $lastCount;
        //构造函数
        function __construct()
        {
            //普通成员变量$lastCount 使用$this 访问,而静态成员变量使用 self 访问
            $this->lastCount=++self::$firstCount;
        }
        //打印 lastCount 数值
        function printLastCount()
        {
            print($this->lastCount);
        }
    }
    //实例化对象
    $obj=new Count();
    $obj->printLastCount();   //执行到这里的时候,程序输出 1
?>
```

5.12 反 射 技 术

生活中，有时候用户需要描述一个自己熟悉的人的行为和特点，有时候已知行为和特点而要用户反过来猜这个人是谁，后者其实是一种反射。在 PHP 中，反射是指用来检

查未知类中的基本信息并对这些基本信息进行操作的机制。反射有什么作用呢？比如要把一个对象中的数据复制到另外一个对象中，复制规则是对象的属性名相同就复制，这可以用反射技术来实现，通过反射技术先动态获取未知类的信息，再判断类属性名是否相同，如果相同就复制。PHP 反射涉及一组类和 API（应用程序接口），借助反射类和 API 可以获取类的结构，创建一个类的实例（不同于用 new 创建），调用一个方法（也不同于常规调用），传递参数和动态调用类的静态方法等。下面的实例利用反射技术如何查看自定义类的信息。

【例 5-12-1】 （代码位置：\5\testReflectionClass. php）

```
<?php
//定义一个自定义类
class MyTestClass{
    public function testFunc($para0='defaultValue0'){
    }
}
//使用反射技术解析类 MyTestClass 的内容
foreach(get_declared_classes()as $class){
    //实例化一个反射类
    $reflectionClass=new ReflectionClass($class);
    //如果该类是自定义类
    if($reflectionClass->isUserDefined()){
        //导出该类信息
        Reflection::export($reflectionClass);
    }
}
?>
```

在上例中，函 get_declared_classes 用来返回由已定义类的名字所组成的数组，获类 ReflectionClass 的成员方法了解自定义类结构，运行结果如下：

```
Class [ class MyTestClass ] { @@C:\PHP\htdocs\ 5\testReflectionClass.php 3- 8 - Constants
[0]{}- Static properties [0] {} - Static methods [0]{}- Properties [0]{}- Methods [1] {
Method [ public method testFunc ] { @@C:\PHP\htdocs\test2_1.php 5 - 7 - Parameters [1] {
Parameter #0 [ $para0='defaultValue0' ]}}}}
```

下面是另一个关于反射的实例，例中程序根据参数来决定要执行哪个函数。

【例 5-12-2】 （代码位置：\5\testReflectionFun. php）

```
<?php
    function test0()
    {
        echo("没有参数");
    }
    function test1($a)
    {
```

```
        echo("一个参数");
    }
    function test2($a,$b)
    {
        echo("二个参数");
    }
    function test3($a,$b,$c)
    {
        echo("三个参数");
    }
    function test()
    {
        $args=func_get_args();                            //获取当前函数的参数
        $argsnum=func_num_args();                         //获取参数的数目
        call_user_func_array('test'.$argsnum,$args);      //根据参数调用相应的函数
    }
    test();
    test(1,2);
?>
```

上例程序最终执行了函数 test0 和 test2,运行结果如下:

```
没有参数二个参数
```

有关反射 API 和类,有兴趣的读者可查阅 PHP 相关手册。

第 6 章 chapter 6

MySQL 数据库开发

6.1 MySQL 基础

6.1.1 MySQL 简介

很多电子商务系统等应用都会用数据库系统来持久化数据，借助数据库系统，应用可以存储和检索数据。常用的数据库系统有 Oracle、SQL Server 和 MySQL 等。MySQL 数据库系统是 PHP 的绝佳搭档。到底什么是 MySQL？MySQL 是一个多用户、多线程的小型关系型数据库管理系统，是一个客户/服务器结构的应用，它由一个服务器守护程序 mysqld、多个不同的客户程序和库组成。MySQL 的开发者为瑞典 MySQL AB 公司。其在 2008 年 1 月被 Sun 公司收购。而 2009 年，Sun 公司又被 Oracle 公司收购。根据 MySQL 的开发情况，可将其分为 Alpha、Beta、Gamma 和 Generally Available（简称 GA）等版本。GA 是指软件的通用版本，是较稳定的版本，也是正式发布的版本。从应用上可将 MySQL 分为社区版（Community Server）、企业版（Enterprise Edition）、集群版（Cluster）和高级集群版（Cluster CGE）等。从历史上来，MySQL 5. x 是较新的版本。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。MySQL 不仅可以通过命令方式管理数据，而且可以借助 MySQL GUI Tools、MySQL-Front 等图形界面客户端软件管理数据，以及通过 phpMyAdmin 等 Web 方式管理数据。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。MySQL 的官方网站的网址是 www. mysql. com。

现在每一个人的生活几乎都离不开数据库，如果没有数据库，很多事情都会变得非常棘手，也许根本无法做得到。银行、大学和图书馆都是严重依赖数据库系统的地方。在 Internet 上，搜索引擎、在线购物系统甚至是普通网站都离不开数据库。与其他大型数据库（例如 Oracle、DB2 和 SQL Server 等）相比，MySQL 有它的不足之处，如规模小、功能有限（MySQL Cluster 的功能和效率都相对比较差）等，但是这丝毫也没有降低它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL 提供的功能已经绰绰有余，而且由于 MySQL 是开放源码软件，因此可以大大降低总体拥有成本。

表 6-1-1　数值类型

类　型	大　小	范围(有符号)	范围(无符号)	用　途
TINYINT	1B	(−128,127)	(0,255)	小整数值
SMALLINT	2B	(−32 768,32 767)	(0,65 535)	大整数值
MEDIUMINT	3B	(−8 388 608,8 388 607)	(0,16 777 215)	大整数值
INT 或 INTEGER	4B	(−2 147 483 648,2 147 483 647)	(0,4 294 967 295)	大整数值
BIGINT	8B	(−9 233 372 036 854 775 808, 9 223 372 036 854 775 807)	(0,18 446 744 073 709 551 615)	极大整数值
FLOAT	4B	(−3.402 823 466 E+38,1.175 494 351E−38),0, (1.175 494 351 E−38,3.402 823 466 351E+38)	0,(1.175 494 351 E−38, 3.402 823 466 E+38)	单精度 浮点数值
DOUBLE	8B	(1.797 693 134 862 315 7 E+308, 2.225 073 858 507 201 4 E−308), 0,(2.225 073 858 507 201 4 E−308, 1.797 693 134 862 315 7 E+308)	0,(2.225 073 858 507 201 4 E−308, 1.797 693 134 862 315 7 E+308)	双精度 浮点数值
DECIMAL	对 DECIMAL(M,D),如果 $M>D$,为 $M+2$,否则为 $D+2$	依赖于 M 和 D 的值	依赖于 M 和 D 的值	小数值

目前 Internet 上流行的网站构架方式是 LAMP(Linux+Apache+MySQL+PHP)，即使用 Linux 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库，PHP 作为服务器端脚本解释器。由于这 4 个软件都是自由或开放源码软件，因此使用这种方式不用花太多的钱就可以快速建立起一个功能强大和稳定的网站系统。

6.1.2 MySQL 数据类型

MySQL 支持大量的数据类型，其可以被分为数字类型、日期和时间类型以及字符串(字符)3 个类型。表 6-1-1、表 6-1-2 和表 6-1-3 给出了 MySQL 所支持的字段数据类型。

表 6-1-2　日期和时间类型

类　型	大小/B	范　　围	格　　式	用　　途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00/ 9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	8	1970-01-01 00:00:00/ 2037 年某时	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

表 6-1-3　字符串类型

类　　型	大小/B	用　　途
CHAR	0～255	定长字符串
VARCHAR	0～255	变长字符串
TINYBLOB	0～255	不超过 255 个字符的二进制字符串
TINYTEXT	0～255	短文本字符串
BLOB	0～65 535	二进制形式的长文本数据
TEXT	0～65 535	长文本数据
MEDIUMBLOB	0～16 777 215	二进制形式的中等长度文本数据
MEDIUMTEXT	0～16 777 215	中等长度文本数据
LONGBLOB	0～4 294 967 295	二进制形式的极大文本数据
LONGTEXT	0～4 294 967 295	极大文本数据

6.1.3 phpMyAdmin 管理软件的安装与配置

phpMyAdmin 是一个使用 PHP 编写的、基于 Web 的 MySQL 客户端程序，是一个图形管理工具，支持简体中文，允许用户使用 Web 浏览器来管理 MySQL 数据库。因为 phpMyAdmin 是基于 Web 的，所以在安装它之前，因此先要保证 Apache Web 服务器的正常运行，并且 Apache Web 服务器要安装相应的 MySQL 和 PHP 扩展插件。phpMyAdmin 的用户界面直观简洁，各部分的内容一目了然。比如要查看现在 MySQL 数据库服务器上都有哪些数据库，直接单击选择主界面上的“数据库”即可。第 1 章介绍

了 PHPNow 套件的安装，由于在正常安装 PHPNow 时，其套件已经绑定安装了 phpMyAdmin，因此，不需要额外再安装 phpMyAdmin 就可以直接使用它了。安装 PHPnow 时，PHPnow 会把 phpMyAdmin 安装在 PHP 安装目录下的子目录 htdocs 中。

6.1.4 phpMyAdmin 的使用

在浏览器地址栏输入 http://127.0.0.1/phpMyAdmin/index.php，就可以访问 phpMyAdmin 应用程序了。其登录界面如图 6-1-1 所示。



图 6-1-1 phpMyAdmin 主界面

输入正确的 MySQL 数据库的用户名和密码，单击"执行"按钮进入应用主界面，可通过这个主界面实现数据库的创建和查询等操作，主界面如图 6-1-2 所示，图的左边是数据库和表，图的右边是相应的操作。要操作数据库或表，可先选择图左边的数据库或表，再选择图右边的按钮等组件进行操作即可。例如，要使用 SQL 语句查询数据库表，可选择"数据库×××"→SQL 菜单命令，在文本框中输入相应的 SQL 命令即可。

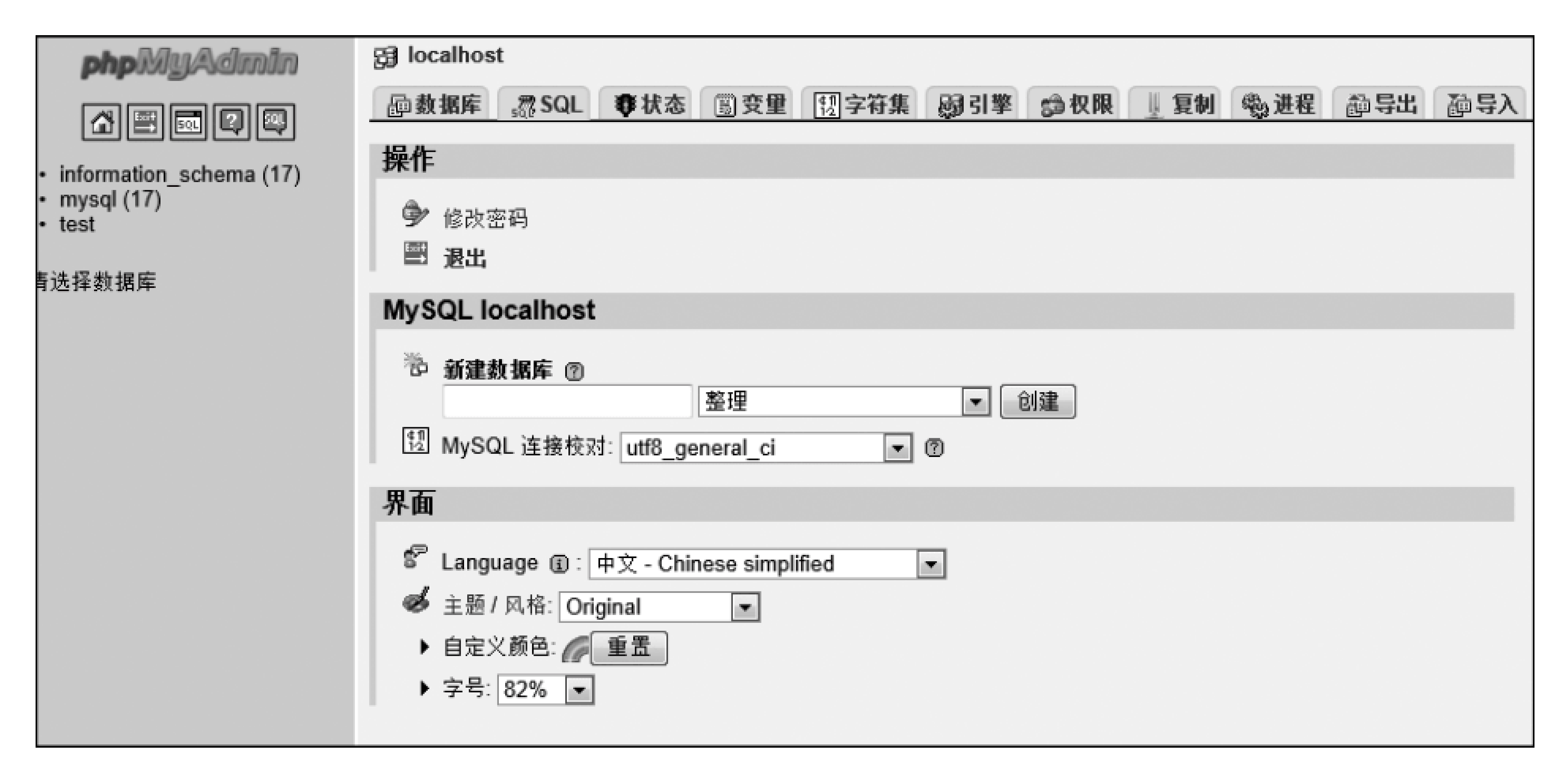


图 6-1-2 phpMyAdmin 主界面

6.1.5 MySQL命令行工具

如果计算机上没有安装 phpMyAdmin 等数据库图形界面管理工具，可以启动 MySQL 命令行工具(PHP 安装目录\MySQL-5.0.90\bin\mysql.exe)使用结构化查询语言 SQL 来操作数据库。在 Windows 命令行界面下输入“mysql -u root -p”命令，接着按回车键，会提示输入密码，此时可输入用户 root 的密码。下面是启动命令行工具的过程，符号 mysql>表示已经登录成功并进入命令行工具，至此可以使用 SQL 语言操作 MySQL 数据库了。

```
C:\PHPnow-1.5.6\MySQL-5.0.90\bin>mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor. Commands end with ;or \g.
Your MySQL connection id is 25
Server version: 5.0.90-community-nt MySQL Community Edition(GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

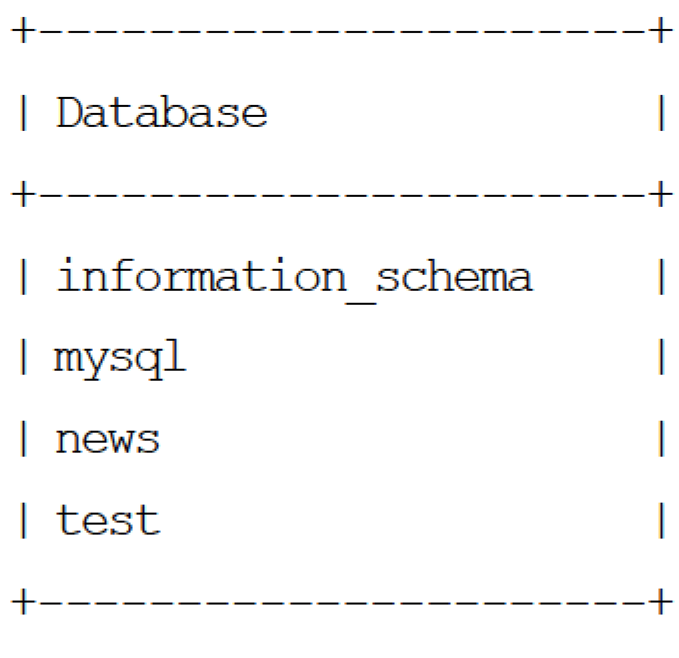
下面以建立、修改和删除数据库 news 和数据库表 news_category 为例讲解使用 SQL 操作 MySQL 数据库。

首先需要创建数据库，可以使用 create database 命令创建数据库。下面是建立一个名为 news 的数据库范例，创建命令和运行结果如下：

```
mysql>create database news;
Query OK,1 row affected(0.00 sec)
```

接下来使用 show databases 命令查看所建数据库是否成功，显示有 news 则表示创建成功，否则创建失败。查看命令和运行结果如下：

```
mysql>show databases;
```



```
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| mysql                |
| news                 |
| test                 |
+----------------------+
4 rows in set(0.00 sec)
```

虽然已经创建了数据库，但还不能使用该库，如果要使用该库，还需要用 use 命令打开数据库。use 命令和运行结果如下：

```
mysql>use news
```

```
Database changed
```

打开数据库后，使用 show tables 查看该库中的数据库表。查看表命令和运行结果如下，可看到该库中没有表。

```
mysql> show tables;
Empty set (0.00 sec)
```

下面使用 create table 命令创建表 news_category，该表有 2 个字段，一个是 id 字段，另一个是 name 字段，创建表命令和运行结果如下：

```
mysql> create table news_category(id int(11)not null,name varchar(60)not nu
ll,primary key(id));
Query OK,0 rows affected(0.13 sec)
```

再次使用 show tables 命令查看数据库中的表，显示结果如下：

```
mysql> show tables;
+----------------+
| Tables_in_news  |
+----------------+
| news_category   |
+----------------+
1 row in set (0.00 sec)
```

从运行结果可知表 news_category 成功创建。如需向表中插入数据，可使用 insert 命令实现，此处向表中插入一条记录，记录的 name 字段值为“新闻 1”，命令和运行结果如下：

```
mysql> insert into news_category(name)values('新闻 1');
Query OK,1 row affected,1 warning(0.05 sec)
```

为了验证数据是否已插入表中，可用 select 命令查看表 news_category 中的数据，select 命令和运行结果如下：

```
mysql> select * from news_category;
+----+-------+
| id | name  |
+----+-------+
| 0  | 新闻 1 |
+----+-------+
1 row in set (0.00 sec)
```

从运行结果可知记录已插到表中，如需修改表记录的字段内容，可使用 update 命令实现，此处把 name 字段值更改为“新闻 2”，update 命令和运行结果如下：

```
mysql> update news_category set name='新闻 2' where name='新闻 1';
Query OK,1 row affected(0.04 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

修改数据之后,可使用 select 命令查看是否修改成功,命令和运行结果如下:

```
mysql> select * from news_category;
+----+-------+
| id | name  |
+----+-------+
| 0  | 新闻 2 |
+----+-------+
1 row in set (0.00 sec)
```

如需删除表中的数据,可使用 delete 命令,此处把表 news_category 中的记录全部删除,delete 命令和运行结果如下:

```
mysql> delete from news_category;
Query OK, 1 row affected (0.01 sec)
```

删除表数据后,可使用 select 命令查看表 news_category 中的记录是否存在,如果不存在则表示成功删除表数据,命令和运行结果如下:

```
mysql> select * from news_category;
Empty set (0.00 sec)
```

如需删除数据库,可使用 drop 命令,drop 命令和运行结果如下:

```
mysql> drop database news;
Query OK, 1 row affected (0.90 sec)
```

可使用 show databases 命令查看数据库 news 是否被删除,命令和运行结果如下:

```
mysql> drop database news;
Query OK, 1 row affected (0.90 sec)

mysql> show databases;
+---------------------+
| Database            |
+---------------------+
| information_schema  |
| mysql               |
| test                |
+---------------------+
3 rows in set (0.00 sec)
```

注:为了能验证后面范例是否正确运行,请把数据库 news 删除。要使用 MySQL 命令行工具操作 MySQL 数据库需要熟悉 SQL 命令,SQL 命令内容很丰富,有关内容请参看其他相关资料。

6.2 PHP 访问 MySQL 数据库

6.2.1 连接和关闭数据库服务器连接

在访问和处理数据库中的数据之前，必须创建到达数据库服务器的连接。函数 mysql_connect 用来打开或重复使用一个到 MySQL 服务器的连接，该函数格式如下：

```
resource mysql_connect([ string $server [,string $username [,string $password]]])
```

如果连接成功则返回一个 MySQL 连接标识，在失败时返回 FALSE。参数 $ server 是指要连接的 MySQL 服务器，默认值是 localhost:3306。参数 $ username 是指用户名，默认值是拥有服务器进程的用户的名称。参数 $ password 是指密码，默认为空。这些参数都是可选的。

数据库访问完毕后需要关闭数据库服务器连接以释放资源。函数 mysql_close 用来关闭 MySQL 连接，其格式如下：

```
bool mysql_close([resource $link_identifier ])
```

参数 $ link_identifier 是指定的 MySQL 数据库服务器连接标识符，它由 mysql_connect 创建。如果没有指定 link_identifier，则关闭上一个打开的连接。通常不需要使用 mysql_close，因为已打开的非持久连接会在脚本执行完毕后自动关闭。关闭数据库连接成功时返回 TRUE，失败时返回 FALSE。下面是连接和关闭数据库的范例。

【例 6-2-1】（代码位置：\6\testMySQL.php）

```
<?php
  $link=mysql_connect('localhost','root','111111');       //创建一个连接
  if(!$link){
  //连接失败则显示错误信息,mysql_error 用来获取出错信息
    die('连接失败: ' . mysql_error());
  }
  echo '连接成功';
  mysql_close($link);                                     //关闭连接
?>
```

运行程序，如果连接成功，则显示"Connected successfully"。如果密码等错误，则提示警告，显示结果如下：

```
Warning: mysql_connect() [function.mysql-connect]: Access denied for user 'root'@'localhost'
(using password: YES) in C:\PHP\htdocs\test2_1.php on line 2 Could not connect: Access denied for
user 'root'@'localhost'(using password: YES)
```

6.2.2 创建数据库

在 PHP 中，可以使用函数 mysql_query 创建数据库，函数格式如下：

```
resource mysql_query(string $query [,resource $link_identifier])
```

函数 mysql_query 会向与连接标识符 $ link_identifier 关联的数据库系统发送一条 SQL 命令。如果没有指定 link_identifier，则使用上一个打开的连接。如果没有打开的连接，那么该函数会尝试无参数调用 mysql_query 来建立一个连接并使用之。参数 $ query 是 SQL 查询串。该函数会自动对记录集进行读取和缓存。

函数 msql_query 仅对 select、show、explain 和 describe 等命令返回一个资源标识符(指向一个结果集)，而这些命令执行不正确时将返回 FALSE。对于其他类型的 SQL 语句，mysql_query 在执行成功时返回 TRUE，出错时返回 FALSE。非 FALSE 的返回值(整数)意味着 SQL 查询语句是合法的并能够被 MySQL 服务器执行。非 FALSE 值并不说明任何有关影响到的或返回的行数。很有可能一条查询执行成功了，但并未影响到或并未返回任何行。下面是创建数据库的范例。

【例 6-2-2】 (代码位置：\6\testCreateDB. php)

```
<?php
  $con=mysql_connect("localhost","root","11111111");
  if(!$con)
  {
    die('连接失败：' . mysql_error());
  }
  if(mysql_query("CREATE DATABASE news",$con))
  {
    echo "创建数据库成功";
  }
  else
  {
    echo "创建数据库出错：" . mysql_error();
  }
  mysql_close($con);
?>
```

程序运行后会和 MySQL 数据库服务器连接，如果连接失败则提示“连接失败”，如果连接成功则发出创建数据库命令。要验证数据库是否创建成功，可使用 MySQL 命令行工具查看数据库是否存在，存在则表示创建成功，否则表示创建失败。

6.2.3 选择数据库

当 PHP 程序与 MySQL 服务器建立连接后，由于数据库服务器中很可能会包含多个数据库，所以需要打开指定的数据库方可对数据库进行操作。在 PHP 中选择数据库使用函数 mysql_select_db 实现，该函数的语法格式如下：

```
boolean mysql_select_db(string db_name[,resource iink_id])
```

参数 db_name 指定要使用的数据库名称，link_id 表示数据库连接资源标识号，通常

是函数 mysql_connect 或 mysql_pconnect 的返回值，如果在函数中没有指定资源标识号，则会试图使用前次连接的资源标识号。函数 mysql_select_db 会与数据库服务器上的一个具体的数据库相连接，连接正确返回 TRUE，连接失败返回 FALSE。

下面是使用函数 mysql_select_db 选择数据库的范例，范例先选择 MySQL 内置的数据库 test，再选择前面创建的数据库 news，最后判断是否选择成功。

【例 6-2-3】 （代码位置：\6\testSelectDB. php）

```
<?php
  //建立连接
  $db=@mysql_connect("localhost","root","111111") or die("!!!连接失败,无法连接到 MySQL
  服务器!");
  echo "已连接到 MySQL 服务器...<br>";
  //选择数据库 test
  if(mysql_select_db("test",$db))
    echo "已选择 test 数据库...<br>";
  else
    echo "数据库 test 选择失败: ".mysql_error();
  //选择数据 news
  if(mysql_select_db("news",$db))
    echo "已选择 news 数据库...<br>";
  else
    echo "数据库 news 选择失败: ".mysql_error();
?>
```

程序运行后，其会连接数据库服务器，连接失败则提示连接失败，成功则先选择数据库 test，再选择库 news，如果选择操作失败会提示选择失败，否则提示连接成功。

6.2.4 创建数据库表

可以使用函数 mysql_query 向服务器发送 SQL 命令创建数据库表。下面是创建数据库表的范例。

【例 6-2-4】 （代码位置：\6\testQuery. php）

```
<?php
  $con=mysql_connect("localhost","root","111111");
  if(!$con)
  {
    die('连接失败: ' . mysql_error());
  }
  mysql_select_db("news",$con);
  $sql="CREATE TABLE news_category
  (
  id int(11)not null,
  name varchar(60),
```

```
  primary key(id)
  )";
  $flag=mysql_query($sql,$con);
  if($flag==false)
    echo "创建表失败";
  else echo "创建表成功";
  mysql_close($con);
?>
```

程序执行后,可以在 MySQL 命令行工具中运行 SQL 命令查看创建数据库是否成功,具体查看方法见本章前述内容。

6.2.5 插入数据

可以使用函数 mysql_query 向服务器发送 SQL 语言往数据库表里插入数据。下面给出向数据库表插入数据的实例。

【例 6-2-5】 (代码位置:\6\testInsert.php)

```
<?php
$con=mysql_connect("localhost","root","111111");
if(!$con)
  {
  die('连接失败: ' . mysql_error());
  }

mysql_select_db("news",$con);

mysql_query("INSERT INTO news_category (id,name)
VALUES('001','新闻 1')");

mysql_query("INSERT INTO news_category (id,name)
VALUES('002','新闻 2')");

mysql_close($con);
?>
```

执行本程序后将在数据库表 news_category 中插入数据,可以在 SQL 命令行程序中通过 select 语句查看数据是否插入成功,具体参看本章前面的内容。

6.2.6 查询数据和显示数据

1. 查询数据

查询 MySQL 数据库首先需要创建一个 SQL 查询语句,然后将该语句传递给执行查询操作的函数即可。在 PHP 中,可以使用函数 mysql_query 执行查询操作。下面是数据库查询的范例。

【例 6-2-6】 (代码位置: \6\testMySQL_query.php)

```
<?php
  //建立连接
  $db=@mysql_connect("localhost","root","11111111")or die("!!!连接失败,无法连接到MySQL
  服务器!");
  echo "已连接到 MySQL 服务器...<br>";
  mysql_select_db("news",$db);
  //执行查询操作
  $result=mysql_query("select * from news_category",$db) or die("<br>查询表 news_
  category 失败!!!");
?>
```

上例程序执行的是 select 命令,如果查询成功将返回一个资源标识号,通过该资源标识号可查看查询到的数据集,但如果数据库 news 中不存在数据表 news_category,那么查询将失败并显示"查询表 news_category 失败"。

2. 显示数据

使用函数 mysql_query 执行 select 语句后将返回指向数据集的资源标识符,如何读出这些数据集并显示出来呢? 函数 mysql_fetch_row 可以实现这个功能。函数 mysql_fetch_row 以数组的形式返回查询结果集中的当前记录行,并在调用后将结果集中的当前行指针下移一行。该函数的语法格式如下:

```
array mysql_fetch_row(resource result_set)
```

参数 result_set 是由函数 mysql_query 返回的资源标识号(标识一个查询结果集)。该函数会从 result_set 中获取当前的数据行,并且以数字索引数组的形式返回。数组的下标从 0 开始,数组中的第 i 个元素的下标为 i−1。下面是选取存储在表 news_category 中所有数据的范例。

【例 6-2-7】 (代码位置: \6\testMySQL_fetch.php)

```
<?php
  $con=mysql_connect("localhost","root","11111111");
  if(!$con)
  {
    die('连接失败: ' . mysql_error());
  }
  mysql_select_db("news",$con);
  $result=mysql_query("SELECT * FROM news_category");
  while($row=mysql_fetch_array($result))
  {
    echo $row['id'] . " " . $row['name'];
    echo "<br />";
  }
  mysql_close($con);
?>
```

上面这个例子先在变量$result 中存放由 mysql_query 函数返回的数据。接下来，使用函数 mysql_fetch_array 以数组的形式从记录集返回第一行记录。以后每次调用函数 mysql_fetch_array 都会返回记录集中的下一行记录。while 语句会通过函数 mysql_fetch_array 循环逐行读出记录集中的所有记录。为了输出每行相应字段的值，这里使用了关联数组变量$row，变量$row['id']和$row['name']分别表示记录中的 id 和 name 字段。运行结果如下：

```
1 新闻 1
2 新闻 2
```

6.2.7 更新数据

下面是更新表数据的范例。

【例 6-2-8】 （代码位置：\6\testUpdate.php）

```
<?php
  $con=mysql_connect("localhost","root","11111111");
  if(!$con)
  {
    die('连接失败：' . mysql_error());
  }
  mysql_select_db("news",$con);
  mysql_query("UPDATE news_category SET name='中国新闻' where name='新闻 1'");
  mysql_close($con);
?>
```

6.2.8 删除数据

下面是删除表数据的实例。

【例 6-2-9】 （代码位置：\6\testDelete.php）

```
<?php
  $con=mysql_connect("localhost","root","11111111");
  if(!$con)
  {
    die('连接失败：' . mysql_error());
  }
  mysql_select_db("news",$con);
  mysql_query("delete from news_category ");
  mysql_close($con);
?>
```

第 7 章



HTML 表单与 PHP 处理

7.1 浏览器/服务器结构工作原理

现在很多人都在网上听歌、看视频和购物，这种网就是万维网，其英文全称是 World Wide Web（简写为 WWW），其中文翻译为"环球信息网"，也常简称其为 Web。Web 采用 B/S（Browser/Server）结构，即浏览器/服务器结构。在这种结构下，用户工作界面是通过 WWW 浏览器来实现的，主要事务逻辑在服务器端实现，很少一部分事务逻辑在浏览器端实现。这样的好处是大大简化了客户端的计算机载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。用户只需要安装浏览器即可浏览页面，不需要知道服务器端使用什么操作系统或服务器端怎么处理浏览器发出的请求，可以方便查看自己想看到的内容。浏览器/服务器结构的工作原理可以分以下几步来理解：

（1）浏览器通过 HTML 表单或超链接等形式向 Web 服务器发送页面请求。

（2）服务器接受浏览器请求。

（3）运行在服务器端的特定 Web 应用程序处理该请求。

（4）Web 应用程序把结果格式化为浏览器能够理解的文档，即通常所说的 HTML 网页。

（5）Web 服务器将 Web 应用程序格式化后的 HTML 网页处理结果返回到浏览器。

（6）浏览器解释 HTML 网页标签，生成相对应的页面并显示出来。

浏览器/服务器结构的工作过程如图 7-1-1 所示。

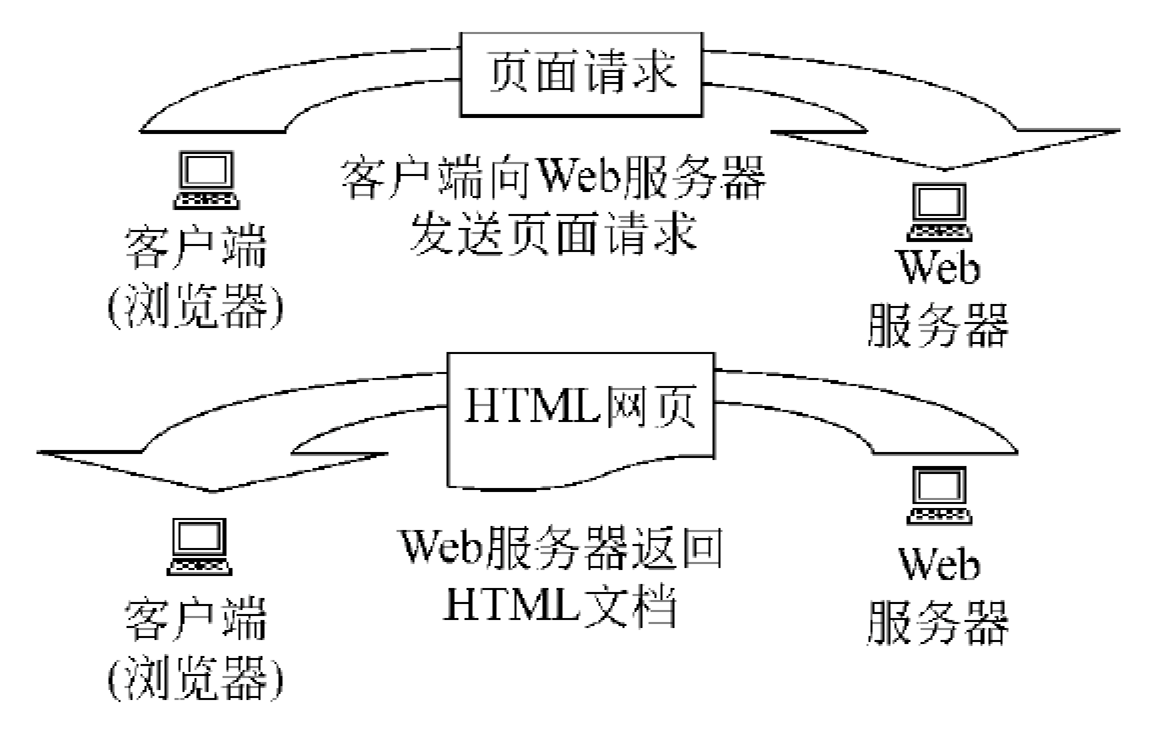


图 7-1-1　客户端/服务器工作过程

7.2 HTML 简介

HTML（Hyper Text Markup Language）是 Internet 上用于编写网页的主要语言。用 HTML 编写的网页文件称为 HTML 文件，HTML 文件以.htm 或.html 为后缀名。

HTML 页面可以使用记事本、Dreamweaver 和 phpDesigner 等软件编写和设计。设计 HTML 语言的目的是为了能把存放在一台计算机中的文本和图形等资源与另一台计算机中的文本和图形等资源方便地联系在一起,形成有机的整体。人们不用考虑具体信息是在当前计算机上还是在网络的其他计算机上,只需使用鼠标单击一个图标,就会马上转到与此图标相关的内容上去,而这些信息可能存放在网络的另一台计算机中。HTML 文本是由 HTML 标签组成的描述性文本,常用的 HTML 标签有文字、图形、动画、声音、表格、链接和表单等标签,通过这些标签能显示出图文并茂的界面,以及实现人机交互、浏览器和服务器交互的目的。HTML 的结构包括头部(head)和主体(body)两大部分,其中头部描述浏览器所需的信息,而主体则包含所要说明的具体内容。下面介绍 HTML 表单及相关标签。

7.3 HTML 表单

HTML 不仅被用来结构化信息,例如标题、段落和列表等,而且 HTML 表单还被用来接收用户的输入信息。HTML 表单在 Web 网页中用来供访问者填写信息,从而能获得用户填写的信息,使网页具有交互的功能。一般是将表单设计在一个 HTML 文档中,当用户填写完信息后做提交(submit)操作,表单的内容就从客户端的浏览器传送到服务器上,经过服务器上的 PHP 等程序处理后,再将用户所需信息传送回客户端的浏览器上,这样网页就具有了交互性。初学者可先侧重学习表单设计与实现,而后再学习表单数据的处理,处理表单数据可参看后面章节。

<form></form>标志对用来定义一个表单,在标志对之间的一切都属于表单的内容。<form>标志具有 action 和 method 属性。action 的值是处理表单的 URL,如<form action="http://www. netsh. com/counter. cgi">,当用户提交表单时,表单信息将被传输给服务器,服务器使用网址 http://www. netsh. com/上名为 counter. cgi 的 CGI 程序来处理表单信息。method 属性用来定义传输表单信息的方式,其值可取 GET 或 POST。GET 这种方式传送的数据量是有所限制的,一般限制在 1KB 以下,同时,这种传输方式不够安全,它适用于安全性度要求不高且数据量较小的应用,例如产品信息查询。POST 方式与 GET 方式相反,POST 方式传送的数据量要比使用 GET 方式大得多,同时,这种传输方式相对安全,适用于安全度要求较高或数量传输量较大的应用,例如用户登录和提交评论。

表单可以包含文本域、列表、按钮和表格等各种形式的输入域,形式丰富的输入域具有较高的用户体验。这些输入域大多可以统一使用<input>标志来定义。一个<input>标志用来定义一个用户输入区,用户可在其中输入信息。此标志必须放在<form></form>标志对之间。<inpu>标志提供了多种类型的输入区域,具体是哪一种类型由 type 属性值来决定。根据不同的 type 属性值,输入区域拥有很多种形式。输入区域可以是文本字段、复选框、单选按钮和提交按钮等,类型由属性 type 的值确定。表单基本语法如下:

```
<form action="*" method="*">
<input type="submit" name="*" value="*">
<input type="text" name="*" value="*">
……
</form>
```

7.3.1 单行文本域

HTML 提供单行文本域用于输入一些简短的信息，如姓名、Email 地址和手机号码等。下面是单行文本域的范例。

【例 7-3-1】 （代码位置：\7\testText.htm）

```
<form>
姓名：<input type="text" name="username" size="20">
</form>
```

姓名：

图 7-3-1 文本域

此例中含有一个输入区域，类型为 text，由 type 属性定义；其名为 username，由 name 属性定义，其值由用户自定义，如存在多个文本框则要避免重名；size＝20 表示输入区域能显示 20 个字符。程序运行效果如图 7-3-1 所示。

7.3.2 密码域

HTML 提供密码域用于用户输入密码。当用户输入密码时，密码域不会显示密码明文，而会用圆点取代密码。下面是密码域的范例。

【例 7-3-2】 （代码位置：\7\testPwd.htm）

```
<form>
用户名：
<input type="text" name="username" size="20" />
<br />
密  码：
<input type="password" name="pwd" size="21"/>
</form>
```

此例中有两个输入区域，一个用于输入用户名，一个用于输入密码，密码输入区域类型为 password，密码输入区域接受用户的信息并以黑圆点显示出来，具有较好的安全性。程序运行效果如图 7-3-2 所示。



图 7-3-2 密码域

7.3.3 单选按钮

当用户从若干给定的选择中选取其一时，就会用到单选按钮。利用 type＝"radio"就会产生单选按钮，单选按钮通常是好几个选项一起摆出来供用户单击选择，因为一次只

能从中选一个选项，故而称单选按钮。单项按钮的每个选项应该包括名字、值和显示内容。下面是单选按钮的范例。

【例 7-3-3】 （代码位置：\7\testRadio.htm）

```
<form>
<input type="radio" name="sex" value="male" />男
<br />
<input type="radio" name="sex" value="female" />女
</form>
```

此例中有一个单选按钮，有两个选项，这两个选项必须同名以表示它们同组，属于同一个单选按钮，此处命名为 sex。前一个选项其值为 male，显示内容为“男”，后一个选项其值为 female，显示内容为“女”。显示内容主要是给用户看，值用来传输给服务器处理。程序运行效果如图 7-3-3 所示。

○ 男
○ 女

图 7-3-3　单选按钮

7.3.4　复选框

当用户需要从若干给定的选择中选取一个或若干选项时，就会用到复选框。利用 type="checkbox"就会产生复选框，复选框通常是好几个选项一起摆出来供用户点选，因一次可以同时选好几个选项，故称复选框。下面是复选框的范例。

【例 7-3-4】 （代码位置：\7\testCheck.htm）

```
<form>
你的喜好：
<input type="checkbox" name="sex" value="movie">电影
<input type="checkbox" name="sex" value="book">看书
</form>
```

此例中有一个复选框，框中有两个选项，前一个选项其值为 movie，显示内容为“电影”，后一个选项其值为 book，显示内容为“看书”。显示内容主要是给用户看，值用来传输给服务器处理。单选按钮和复选框的显示内容可以是图像，通过＜img＞标签设置即可。程序运行效果如图 7-3-4 所示。

你的喜好：　□电影　□看书

图 7-3-4　复选框

7.3.5　重置按钮和提交按钮

通常用户填完信息之后，都会有一个提交按钮以及清除重写的按钮，提交按钮用于把用户填写的信息传输给服务器处理，重置按钮用于清除输入区域已填写的内容，分别利用 type="submit"及 type="reset"来产生。下面是重置按钮和提交按钮的范例。

【例 7-3-5】 （代码位置：\7\testResetSubmit.htm）

```
<form name="input" action="html_form_action.php" method="get">
用户名：
```

```
<input type="text" name="user" />
<input type="reset" value="重置" />
<input type="submit" value="提交" />
</form>
```

此例中，用户已经输入“宝贝”，若用户单击“重置”按钮时，“宝贝”两字即消失；若用户单击“提交”按钮，浏览器将把“宝贝”两字传输给服务器中的 html_form_action. php 程序处理。程序运行结果如图 7-3-5 所示。

图 7-3-5　重置和提交按钮

7.3.6　按钮

有时，单击页面的按钮并不需要把数据提交给服务器处理，而是给浏览器的 JavaScript 脚本实现特效，例如，单击按钮开启或关闭时钟显示。可以利用 type="button"来产生一个按钮。下面是按钮的范例。

【例 7-3-6】 （代码位置：\7\testButton. htm）

```
<form>
关闭按钮：<input type="button" name="ok" value="关闭" onclick="alert('测试按钮')" />
</form>
```

该程序可以不部署到 Web 服务器而直接使用浏览器打开，打开程序后，用户单击“关闭”按钮将跳出“测试按钮”对话框。因为程序并未部署到服务器中，故跳出对话框的只能是浏览器。程序运行效果如图 7-3-6 所示。

图 7-3-6　按钮

7.3.7　多行文本域

有时候想让用户输入量比较大的文字，例如，在论坛系统中输入长篇评论，此时，单行文本域就显得不敷使用。可以利用<textarea></textarea>来产生一个可以输入大量文字的输入域，夹在两个标签中的文字会出现在输入域框中，可作为预设文字，属性 rows 指定可显示的文本行数，属性 cols 指定每行显示的文本字数。文本域基本语法如下：

```
<textarea name=*  rows=*  cols=* >…</textarea>
```

下面是多行文本输入域的范例。

【例 7-3-7】 （代码位置：\7\testTextarea. htm）

```
<html>
    <body>
        <form>
        请发表您的看法：<br>
        <textarea name="talk" cols="25" rows="5"></textarea>
        </form>
    </body>
```

```
</html>
```

该例中，文本域能显示 3 行文字，一行显示 25 个字，需要注意的是，文本域由<textarea>标签而不是由<input>定义。程序运行效果如图 7-3-7 所示。

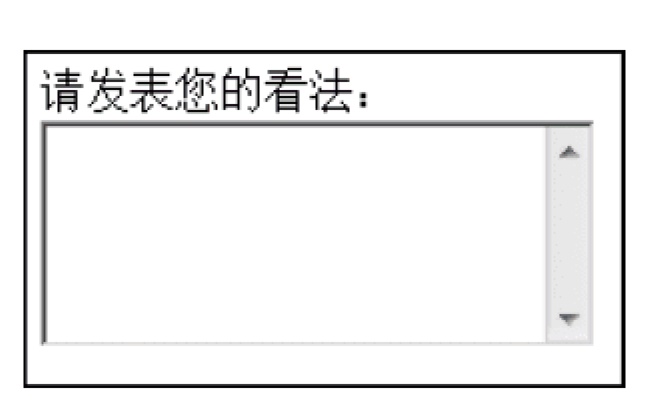


图 7-3-7 多行文本域

7.3.8 列表

有时，需要以有限的区域来显示多个选项供用户选择，可使用 HTML 的下拉列表实现。可利用<select>标签产生一个下拉列表，另外，还需要配合<option>标签来产生选项。<select></select>标志对用来创建一个下拉列表框或可以复选的列表框。<select>具有 multiple、name 和 size 属性。multiple 属性不用赋值，直接加入标志中即可使用，加入了此属性后列表框就成了可多选的了；name 是此列表框的名字，它与前面讲的 name 属性作用是一样的；size 属性用来设置列表的高度，默认值为 1，若没有设置(加入)multiple 属性，显示的将是一个弹出式的列表框。<option>标签用来指定列表框中的一个选项，它放在<select></select>标志对之间。option 标志具有 selected 和 value 属性，selected 用来指定默认的选项，value 属性用来给<option>指定的那一个选项赋值，这个值是要传送到服务器上的，服务器正是通过调用<select>区域的名字的 value 属性来获得该区域选中的数据项的。列表基本语法如下：

```
<select name= *  size= *  multiple>
<option value= *  selected>选项 1</option>
<option value= * >选项 2 </option>
<option value= * >选项 3 </option>
……
</select>
```

下面是下拉列表的范例。

【例 7-3-8】 （代码位置：\7\testList.htm）

```
<html>
    <body>
        <form action="man.php" method="post">
            <p>请选择最喜欢的男歌星：
            <select name="gx1" size="1">
                <option value="ldh">刘德华
                <option value="zhxy" selected>张学友
                <option value="gfch">郭富城
                <option value="lm">黎明
            </select>
        </form>请选择最喜欢的男歌星：
        刘德华 张学友 郭富城 黎明
        <form action="woman.php" method="post">
            <p>请选择最喜欢的女歌星：
```

```
            <select name="gx2" multiple size="4">
                <option value="zhmy">张曼玉
                <option value="wf" selected>王菲
                <option value="tzh">田震
                <option value="ny">那英
            </select>
        </form>请选择最喜欢的女歌星：
        张曼玉 王菲 田震 那英
    </body>
</html>
```

此例中有两个下拉列表，每个下拉列表都有4个选项。前面一个下拉列表只显示1个选项，用户单击下拉列表将弹出4个选项，只允许用户选择一个选项。后面一个下拉列表同时显示4个选项，默认选择第二个选项“王菲”，用户可以选择多个选项。程序运行效果如图7-3-8所示。

图 7-3-8 列表

7.4 PHP获取表单数据

PHP提供了3个全局变量用来获取表单数据：$_GET、$_POST和$_REQUEST。PHP提供变量$_GET用于收集通过method="get"传输的表单数据。变量$_GET是一个数组，内容是由HTTP GET方法发送的变量名称和值。用GET方法传输的表单数据对任何人都是可见的(会显示在浏览器的地址栏)，并且对发送的信息量也有限制。下面是注册的范例，范例要求注册时要输入年龄，年龄小于60不允许注册。本例由Register.htm和Register.php两个文件组成，Register.htm含有HTML表单，用户需在此表单中填写年龄，Register.php用来获取表单数据并判断年龄是否合法。

【例 7-4-1】 (代码位置：\7\Register 目录)

文件Register.htm内容如下：

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <meta name="author" content="WWW.DOWNG.COM"/>
    <title>老年网</title>
```

```
</head>
<body>
<form action="register.php" method="get">
注册前请输入您的年龄:<input type="text" name="age"/>
<input type="submit" value="下一步"/>
</form>
</body>
</html>
```

文件 register. php 内容如下:

```
<?php
  $age=$_GET["age"];
  if($age<60)
    echo "你的年龄小于 60 岁,不符合注册条件";
  else echo "符合注册条件";
?>
```

在 PHP 中,预定义的变量 $_POST 用来收集通过 method="post"发送的表单数据。变量 $_POST 是一个数组,内容是由 HTTP POST 方法发送的变量名称和值。通过 POST 方式从表单发送的信息,对其他人来说是不可见的,在信息发送数量上也没有限制。然而,POST 方法还是有个最大 8MB 的容量限制的,这是默认设置(可以在 PHP 安装目录下的 php-apache2handler. ini 文件中修改 post_max_size 中的设置)。下面是登录范例,登录用户名和密码都为 test。本例由文件 login. htm 和 login. php 组件组成,login. htm 要求用户输入用户名和密码,login. php 会获取表单数据并判断用户名和密码是否都为 test,根据结果显示"合法用户"或"非法用户"。

【例 7-4-2】 (代码位置:\7\Login 目录)

文件 login. htm 内容如下:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <meta name="author" content="WWW.DOWNG.COM"/>
    <title>老年网</title>
</head>
<body>
<form action="login.php" method="post">
用户名:<input type="text" name="uid"/>
密   码:<input type="password" name="pwd"/>
<input type="submit" value="登录"/>
</form>
</body>
</html>
```

文件 login. php 内容如下:

```
<?php
  $uid=$_POST["uid"];
  $pwd=$_POST["pwd"];
  if(($uid=="test")&&($pwd=="test"))
    echo "合法用户";
  else echo "非法用户";
?>
```

PHP 的变量 $_REQUEST 可用来取得通过 GET 和 POST 方法发送的表单数据。其用法和变量 $_POST、$_GET 一样，只是它不再区分表单的方法。读者可自行把上面两例中的 $_GET 和 $_POST 变量改成 $_REQUEST 变量，观察结果。

7.5 问卷调查系统

本节通过问卷调查系统实例向读者介绍 PHP 如何处理表单的各种数据，包括单选和多选数据。问卷调查系统是一种要求用户对给定问题进行选择，并对用户的选项进行收集、统计和显示的软件。它由问卷界面和逻辑处理两部分组成，问卷界面向用户显示问题和选项，用户通过界面选择选项，并把问卷提交给逻辑部分处理，逻辑部分用来收集用户的选项并把选项值显示给用户供用户核实。下面是 PHP 表单处理范例，用于问卷调查，由文件 answer.htm 和 answer.php 组成，前者用于显示界面，后者用于业务逻辑处理。

【例 7-5-1】 （代码位置：\7\Answer 目录）

文件 answer.htm 内容如下：

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <meta name="author" content="WWW.DOWNG.COM"/>
    <title>老年网</title>
</head>
<body>
<form action="answer.php" method="post">
用户名<br/>
<input type="text" name="uid"/><br/>
年  龄<br/>
<select name="age">
  <option>0-19</option>
  <option selected="true">20-60</option>
  <option>60 以上</option>
</select>
<br/>
自我介绍：<br/>
```

```
<textarea name="intro" rows="3" cols="40"></textarea>
<br/>
爱好:<br/>
<input type="radio" name="favo" value="singing"/>唱歌
<input type="radio" name="favo" value="sports"/>体育
<input type="radio" name="favo" value="game"/>游戏
<input type="radio" name="favo" value="study"/>学习
<br/>
精通语言:<br/>
<input type="checkbox" name="language[]" value="C"/>C
<input type="checkbox" name="language[]" value="C++"/>C++
<input type="checkbox" name="language[]" value="JAVA"/>JAVA
<br/>
常用软件:<br/>
<select name="tools[]" size="3" multiple="true" >
  <option value="Office">Office</option>
  <option value="VC">VC</option>
  <option value="Eclipse">Eclipse</option>
</select>
<br/>
<input type="submit" value="提交"/>
</form>
</body>
</html>
```

值得注意的是,单选按钮元素的 name 属性值要相同,例如 favo,而复选框和支持多选的列表元素的 name 属性值不仅其值要相同,而且尾部要有中括号符号“[]”,例如 language[]和 tools[]。在界面运行后输入内容,其界面和所输入的内容如图 7-5-1 所示。

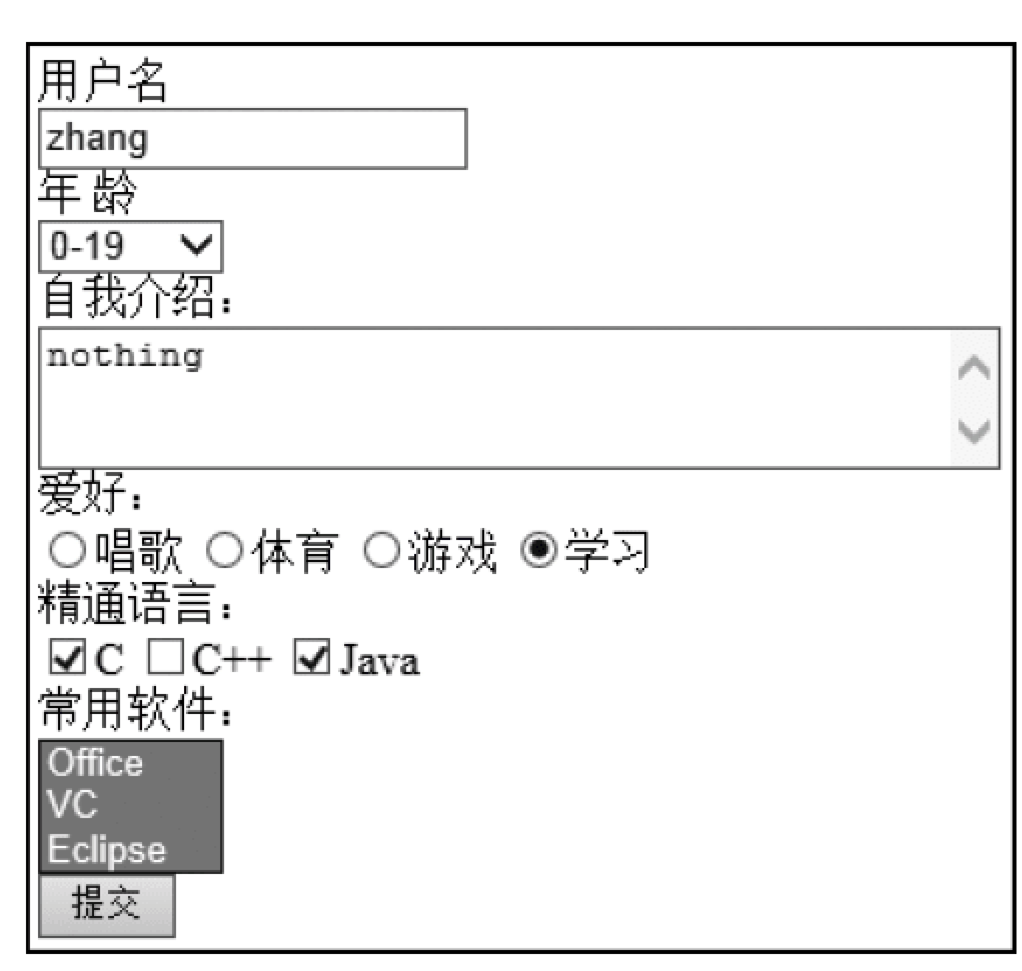


图 7-5-1 问卷问答系统界面

按图中内容输入数据后,单击“提交”按钮,系统将运行程序 answer.php 处理表单数据,其代码如下。

文件 answer.php 内容如下:

```
<?php
  //获取表单数据
  $uid=$_REQUEST["uid"];
  $age=$_REQUEST["age"];
  $intro=$_REQUEST["intro"];
  $fave=$_REQUEST["favo"];
  $language=$_REQUEST["language"];
```

```
  $tools=$_REQUEST["tools"];
  //输出表单数据
  echo "你输入的用户名：$uid <br/>";
  echo "你输入的年龄：$age <br/>";
  echo "你输入的自我介绍：<br/>$intro <br/>";
  echo "你输入的爱好：$favo <br/>";
  echo "你输入的开发语言：<br/>";
  //输出表单中的多选数据
  foreach($language as $value)
    echo "$value <br/>";
  echo "你输入的工具：<br>";
  foreach($tools as $value1)
    echo "$value1 <br/>";
?>
```

上述代码被正确执行后将输出如下内容：

```
你输入的用户名：zhang
你输入的年龄：0-19
你输入的自我介绍：
nothing
你输入的爱好：study
你输入的开发语言：
C
JAVA
你输入的工具：
Office
VC
Eclipse
```

第 8 章



XML 技术

8.1 XML 简介

XML(Extensible Markup Language)的中文名称是可扩展标记语言,它与 HTML 一样,都是 SGML(Standard Generalized Markup Language,标准通用标记语言)的一个子集。在 20 世纪 80 年代早期,IBM 公司提出在各文档之间共享一些相似的属性,例如字体大小。IBM 公司设计了一种文档系统,通过在文档中添加标记来标识文档中的各种元素,经过若干年的发展,1984 年国际标准化组织开始对这种文档系统进行讨论,并于 1986 年正式发布了为生成标准化文档而定义的标记语言标准(标准号是 ISO 8879),它就是 SGML。尽管 XML 和 HTML 同属 SGML 的子集,但它们却是两种不同用途的标记语言。下面通过将 XML 和 HTML 加以比较来认识 XML。

(1) XML 是一种元标记语言,所谓"元标记"就是开发者可以根据自己的需要定义自己的标记,比如开发者可以定义<book>和<name>等各种不同的标记,任何满足 XML 命名规则的名称都可以被定义为标记,这使得应用程序具有较好的数据扩展性。HTML 是一种预定义标记语言,它只认识诸如<html>和<p>等已经定义的标记,而不能识别用户自己定义的标记。

(2) 不是所有的 HTML 标记都需要成对出现,而 XML 则要求所有的标记必须成对出现;HTML 标记不区分大小写,而 XML 则大小敏感,即区分大小写。

(3) XML 是被设计用来描述数据的,重点是:什么是数据,如何存放数据。HTML 是被设计用来显示数据的,重点是:显示数据以及如何显示数据。HTML 是与显示信息相关的,XML 则是与描述信息相关的。例如,HTML 中的<H1>标记可同时表示 20 磅的 Helvetica 字体的粗体、第一级标题和页面标题,XML 中的<bookname>可能表示的是书名。

(4) XML 并不是 HTML 的替代品,理解这一点非常重要。在将来的网页开发中,XML 将用来描述和存储数据,而 HTML 则是用来格式化和显示数据的。

下面用 HTML 和 XML 描述同一首歌曲,通过两者的比较来认识 XML 的优势。用 HTML 描述的歌曲如下:

```
<dt>Hot Cop
<dd>by Jacques Morali Henri Belolo and Victor Willis
<ul>
<li>Producer: Jacques Morali
<li>Publisher: PolyGram Records
<li>Length: 6:20
<li>Written: 978
<li>Artist: Village People
</ul>
```

这首歌用预定义的<dt>、<dd>和<ul>等标志来描述歌名、歌作者和其他信息。但是事实上这些标志没有一个是与音乐有关的。而在XML中，这首歌的数据可能写成如下标记：

```
<SONG>
  <TITLE>Hot Cop</TITLE>
  <COMPOSER>Jacques Morali</COMPOSER>
  <COMPOSER>Henri Belolo</COMPOSER>
  <COMPOSER>Victor Willis</COMPOSER>
  <PRODUCER>Jacques Morali</PRODUCER>
  <PUBLISHER>PolyGram Records</PUBLISHER>
  <LENGTH>6:20</LENGTH>
  <YEAR>978</YEAR>
  <ARTIST>Village People</ARTIST>
</SONG>
```

在这个清单中没有使用通用的标记，如<dt>和<li>，而是使用了具有意义的标记，如<SONG>、<TITLE>、<COMPOSER>和<YEAR>等。这种用法具有许多优点，包括源码易于被人阅读，使人能够看出作者的含义。XML标记还使非人类的自动机器人易于找出文档中的所有歌曲。在HTML中，机器人只能告诉使用者这个元素是dt。机器人不能确定dt到底代表一首歌的题目还是定义，抑或只是一些设计者喜爱的缩进文本格式。

XML是Internet环境中跨平台的和依赖于内容的技术，是当前处理结构化文档信息的有力工具。XML是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然XML占用的空间比二进制数据要占用更多的空间，但XML极其简单，易于掌握和使用。XML的用途很多，可以用来存储数据和做数据交换，下面是XML在应用程序中较常用的用法。

（1）数据交换。XML可用于数据交换，主要是因为XML表示的信息独立于平台，这里的平台既可以理解为不同的应用程序，也可以理解为不同的操作系统，利用它，Microsoft的word文档可以和Adobe的Acrobat交换信息，可以和数据库交换信息。用XML在应用程序和公司之间作数据交换已不是什么秘密了，毫无疑问应被列为数据交换方式中的首位。

(2) Web服务。Web服务是最令人激动的革命之一,它让使用不同系统和不同编程语言的人们能够相互交流和分享数据,其基础在于Web服务器用XML在系统之间交换数据。交换数据通常用XML标记,能使协议规范一致,比如在简单对象处理协议(Simple Object Access Protocol,SOAP)平台上。SOAP可以在用不同编程语言构造的对象之间传递消息。这意味着一个C#对象能够与一个Java对象进行通信。这种通信甚至可以发生在运行于不同操作系统上的对象之间。DCOM、CORBA或Java RMI只能在紧密耦合的对象之间传递消息,SOAP则可在松耦合对象之间传递消息。

(3) 内容管理。XML只用元素和属性来描述数据,而不提供数据的显示方法。这样,XML就提供了一个优秀的方法来标记独立于平台和语言的内容。使用像XSLT这样的语言能够轻易地将XML文件转换成各种格式文件,比如HTML、WML和PDF和ED等。XML具有的能够运行于不同系统平台之间和转换成不同格式目标文件的能力使得它成为内容管理应用系统中的优秀选择。

(4) 配置文件。许多应用程序都将配置数据存储在各种文件里,比如.INI文件。虽然这样的文件格式已经使用多年并一直很好用,但是XML还是以更为优秀的方式为应用程序标记配置数据。配置数据标记为XML格式,更具有可读性,更能方便地集成到应用系统中去。使用XML配置文件的应用程序能够方便地处理所需数据,不用像其他应用那样要经过重新编译才能修改和维护应用系统。

8.2 SimpleXML技术

要处理XML文件,有两种传统处理技术:SAX和DOM。DOM是XML Document Object Model的缩写,中文翻译为XML文档对象模型。当XML文档比较复杂,或者需要随机处理XML文档中数据的时候,DOM是XML文档处理的首选。DOM虽然是W3C的标准,提供了标准的解析方式,但它的解析效率一直不尽如人意,因为使用DOM解析XML时,解析器读入整个文档并构建一个驻留内存的树结构(节点树),然后程序代码才可以使用DOM的标准接口来操作这个树结构。但大多数情况,只需对文档的部分内容感兴趣,根本就不用先解析整个文档,并且从节点树的根节点来索引一些需要的数据也是非常耗时的。SAX是Simple API for XML的缩写,它并不是由W3C官方所提出的标准,可以说是"民间"的事实标准。实际上,它是一种社区的讨论产物。虽然如此,在XML中对SAX的应用丝毫不比DOM少,几乎所有的XML解析器都会支持它。与DOM比较而言,SAX是一种轻量级的方法。SAX在概念上与DOM完全不同。首先,不同于DOM的文档驱动,它是事件驱动的,也就是说,它并不需要读入整个文档。所谓事件驱动,是指一种基于回调(callback)机制的程序运行方法。其次,它在读入XML文档的过程中就进行解析,也就是说读入文档的过程和解析的过程是同时进行的,这和DOM区别很大。另外,SAX在解析开始之前,需要注册一个事件监听器,当在解析过程中读到合适的内容,就会抛出相应的事件给事件监听器,并把这个事件的处理权交给相应的方法进行响应。

除了DOM和SAX这两种传统的方法外,PHP 5还推出了一套新的XML处理函

数，即 SimpleXML。顾名思义，SimpleXML 本身小巧精干，只提供了少量的几个方法函数，但用它们来处理 XML 文件功能却非常强大，操作也非常简单。

首先，SimpleXML 提供了可以直接从 XML 文档、字符串和 DOM 对象上构造出 SimpleXMLElement 对象的简单的函数；其次，SimpleXMLElement 对象提供了可以进行属性、子节点和 XPath 操作的简单的成员方法；然而，SimpleXML 最简单的地方是，它提供了使用标准对象的属性和对象迭代器进行节点操作的方法，这一处理思路使得用 PHP 对 XML 文档的处理得到了极大简化，使得 XML 文档具有了面向对象的特点。和 DOM 等技术相比，SimpleXML 仅仅用几行代码就可以从元素中读取文本数据。表 8-2-1 给出了 SimpleXML 函数。

表 8-2-1 SimpleXML 函数

函　数	描　述
__construct	创建一个新的 SimpleXMLElement 对象
addAttribute	给 SimpleXML 元素添加一个属性
addChild	给 SimpleXML 元素添加一个子元素
asXML	从 SimpleXML 元素获取 XML 字符串
attributes	获取 SimpleXML 元素的属性
children	获取指定节点的子
getDocNamespaces	获取 XML 文档的命名空间
getName	获取 SimpleXML 元素的名称
getNamespaces	从 XML 数据获取命名空间
registerXPathNamespace	为下一次 XPath 查询创建命名空间语境
simplexml_import_dom	从 DOM 节点获取 SimpleXMLElement 对象
simplexml_load_file	从 XML 文档获取 SimpleXMLElement 对象
simplexml_load_string	从 XML 字符串获取 SimpleXMLElement 对象
xpath	对 XML 数据运行 XPath 查询

8.2.1 建立 XML 文件和读取 XML 节点

下面是使用 SimpleXML 建立 XML 文件和读取 XML 内容节点的范例。

【例 8-2-1】 （代码位置：\8\testCreateXML. php）

```
<?php
  $xmlstring=<<<XMLCONTENT
<?xml version="1.0" encoding="utf-8"?>
<note>
<to>George</to>
<from>John</from>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
</note>
XMLCONTENT;
```

```
    $xml=simplexml_load_string($xmlstring);
    echo $xml->to[0];
    echo "<br/>";
    echo $xml->from[0];
    echo "<br/>";
    echo $xml->heading[0];
    echo "<br/>";
    echo $xml->body;
    $xml->asXML("ev.xml");
?>
```

例中使用函数 simplexml_load_string 把包含在 XMLCONTENT 中的 XML 内容转换成对象，用变量 $ xml 引用该对象，并通过"对象－>属性"形式对 XML 内容进行访问，例如 $ xml－> to[0]。程序运行后会生成 ev. xml 文件，文件内容是包含在 XMLCONTENT 中的内容，可在程序所在目录下打开文件 ev. xml 查验内容是否正确。同时程序还会在浏览器中输出如下内容：

```
George
John
Reminder
Don't forget the meeting!
```

8.2.2 新添节点和属性

下面是使用 SimpleXML 新添 XML 节点和属性的范例。

【例 8-2-2】 （代码位置：\8\testNodeAttribute. php）

```
<?php
  $xml=simplexml_load_file('ev.xml');
  $app=$xml->addChild('date');
  $app->addAttribute('year',"2012");
  $app->addChild('month','10');
  $app->addChild('day','1');
  $xml->asXML('ev.xml');
?>
```

例中先加载之前创建的文件 ev. xml，加载后会自动把文件中的 XML 内容转成对象，对象通过变量 $ xml 引用；然后在根元素下新添 date 元素，元素中有属性 year，属性值为 2012；再然后在 date 元素下新添两个子元素 month 和 day，元素值分别为 10 和 1；最后使用 asXML 函数把 XML 内容写回到文件 ev. xml，新添节点和属性后的 ev. xml 文件内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<note>
<to>George</to>
```

```
<from>John</from>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
<date year="2012"><month>10</month><day>1</day></date></note>
```

8.2.3 修改节点

要修改XML节点,可先通过xpath搜寻到指定的节点,再对节点内容进行修改,其范例如下所示。

【例8-2-3】 (代码位置:\8\testModifyNode.php)

```
<?php
  $xml=simplexml_load_file('ev.xml');
  $xg=$xml->xpath("/note/date[@year='2012']");
  $xg[0]->month="1";
  $xg[0]->day="20";
  $xml->asXML('ev.xml');
?>
```

程序运行后,先加载前面范例的XML文件ev.xml,然后SimpleXML借用xpath定位属性year值为2012的/note/date节点。上面的范例新添节点的内容被修改,month和day元素值被修改为1和20,修改后的内容如下:

```
<?xml version="1.0" encoding="utf-8"?>
<note>
<to>George</to>
<from>John</from>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
<date year="2012"><month>1</month><day>20</day></date></note>
```

8.2.4 删除节点

删除XML节点用函数unset实现,但要删除指定节点比较麻烦,只能使用节点的索引值来实现。下面是删除节点的范例。

【例8-2-4】 (代码位置:\8\testRemoveNode.php)

```
<?php
  $xml=simplexml_load_file('ev.xml');
  unset($xml->date);
  $xml->asXML('ev.xml');
?>
```

程序运行后将删除XML文件ev.xml中的date元素节点。

8.3 Web服务

8.3.1 服务简介

Web服务是当前的主流技术之一,其基础是XML技术。想要理解Web服务,必须先理解什么是服务。传统上,把计算机后台程序(daemon)提供的功能称为服务(service)。比如,让一个杀毒软件在后台运行,它会自动监控系统,那么这种自动监控就是一个服务。通俗地说,服务就是计算机可以提供的某一种功能。根据来源的不同,服务又可以分成两种:一种是本地服务(使用同一台计算机提供的服务,不需要网络);另一种是网络服务(使用另一台计算机提供的服务,必须通过网络才能完成)。

举例来说,现在有一批图片,需要把它们的大小缩小一半。可以把“缩放图片”看成是一种服务。可以使用本地服务,在自己计算机上用软件缩小图片;也可以使用网络服务,将图片上传到某个网站,让服务器缩小图片,完成后再通过网络送回来。这就好比一件事用户可以自己做,也可以交给另一个人去做。肚子饿了,用户可以自己做饭,也可以打电话去订一份比萨,让店家替用户做好送上门。

网络服务的本质就是通过网络调用其他网站的资源。举例来说,用户打算做一个“微博图片墙”,它能动态显示关于微博的最新图片。但是,所有的图片都不是储存在用户的服务器上,而是来自flickr.com。用户只是发出一个动态请求,要求flickr.com向用户提供图片。这种情况下,flickr.com提供的就是一种Web Services。如果用户把图片都存放在本地服务器,不调用flickr.com,那么用户就是在使用“本地服务”。

Web服务是一种网络服务,它能网站使用其他网站的资源,比如在网页上显示天气、地图和twitter上的最新动态,等等。

8.3.2 Web服务概念

Web服务也称Web Services,简单说,Web Services就是一些服务器站点开放供客户访问的服务,也可以是自己开发的服务(也即是实现某些功能的方法),通过URL为服务指定某一个服务名,客户发出请求,站点里的这个服务(方法)接到客户的请求,根据传过来的参数,做一些处理,然后把处理后的结果以XML形式返回来给客户,客户程序就解析这些XML数据,然后显示出来或做其他操作。举个例子:很多大的站点提供天气预报的Web Services,应用只要向其发出服务请求,它就会返回天气预报的数据,然后应用把结果显示在自己的界面上。下面是调用Web服务的范例,例中调用天气预报服务查看三亚市的天气信息,具体代码如下所示。

【例8-3-1】 (代码位置:\8\testWebServices.php)

```
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
</head>
```

```
<?php
  $client=new SoapClient('http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?
  wsdl');                        //查询 wsdl
  $parm=array('theCityCode'=>'三亚','theUserID'=>'');
  $result=$client->getWeather($parm);   //调用 getWeather 服务
  print_r($result);
?>
```

把上述代码的文件以UTF-8编码格式保存并部署到自己的服务器中，在浏览器中浏览该程序，程序会向网站http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?wsdl发送天气预报服务请求，该网站会根据请求生成天气预报数据并返回给程序。浏览器将显示如下结果：

```
stdClass Object([getWeatherResult]=>stdClass Object([string]=>Array([0]=>海南 三亚 [1]=>三亚 [2]=>2471 [3]=>2012/08/29 09:51:09 [4]=>今日天气实况：气温：27℃；风向/风力：西风 1级；湿度：86%[5]=>空气质量：良；紫外线强度：中等 [6]=>穿衣指数：天气炎热，建议着短衫、短裙、短裤、薄型T恤衫、敞领短袖棉衫等清凉夏季服装。感冒指数：暂无。运动指数：天气较好，较适宜于户内运动，因气温较高，适当降低运动强度，风力较强，在户外运动注意防风。洗车指数：较不宜洗车，未来一天无雨，风力较大，如果执意擦洗汽车，要做好蒙上污垢的心理准备。晾晒指数：天气不错，适宜晾晒。赶紧把久未见阳光的衣物搬出来吸收一下太阳的味道吧！旅游指数：白云飘飘，温度稍高，幸好风稍大，会缓解稍热的天气。适宜旅游，可不要错过机会哟！路况指数：天气较好，路面比较干燥，路况较好。舒适度指数：天气较好，但空气湿度较大，您会感到有些闷热，不很舒适。空气污染指数：气象条件有利于空气污染物稀释、扩散和清除，可在室外正常活动。紫外线指数：属中等强度紫外线辐射天气，外出时建议涂擦SPF高于15、PA+的防晒护肤品，戴帽子、太阳镜。[7]=>8月29日 多云 [8]=>27℃/32℃ [9]=>西南风4-5级转3-4级 [10]=>1.gif [11]=>1.gif [12]=>8月30日 多云转雷阵雨 [13]=>26℃/32℃ [14]=>西南风3-4级 [15]=>1.gif [16]=>4.gif [17]=>8月31日 雷阵雨转阵雨 [18]=>26℃/31℃ [19]=>西南风3-4级转东南风3-4级 [20]=>4.gif [21]=>3.gif [22]=>9月1日 阵雨转小雨 [23]=>26℃/31℃ [24]=>东南风3-4级 [25]=>3.gif [26]=>7.gif [27]=>9月2日 小雨转小到中雨 [28]=>25℃/31℃ [29]=>东南风3-4级 [30]=>7.gif [31]=>21.gif)))
```

8.3.3 Web服务工作原理

Web Services允许客户端和服务端之间交换XML文档。客户端和服务器端的通信是松耦合的，即Web Services不关心客户端和服务器端所使用的操作系统、对象模型和编程语言等。只要Web Services和使用Web Services的应用程序都能够遵守适当协议标准发送和接收消息即可。Web服务工作原理可用图8-3-1表示。

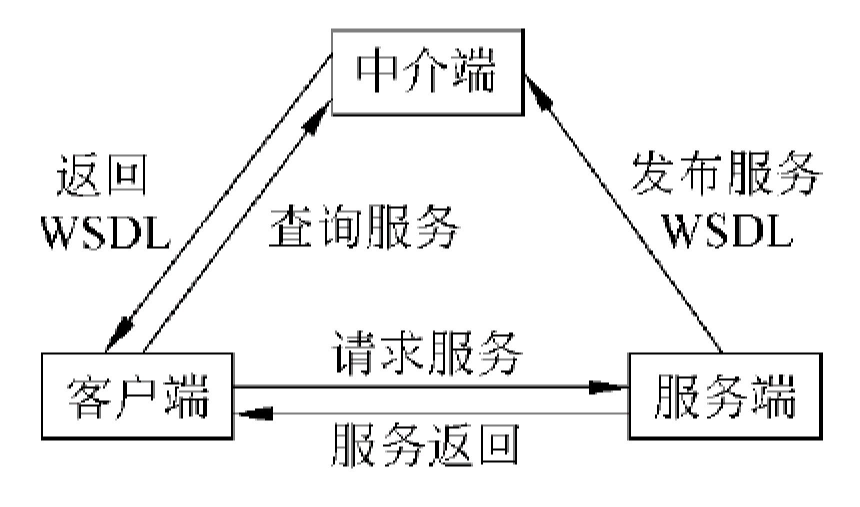


图 8-3-1 Web服务工作原理

服务端首先用一套标准的方法向外界描述它所提供的服务内容，这套标准的方法就是WSDL(Web Services Description Language，Web Services描述语言)。然后服务器端将WSDL文件放在一个公共的网站供客户端查询服务内容。最后客户端通过这个公共

的网站查询服务内容,并以一种标准的协议来访问服务器端调用此服务,这个标准的协议就是 SOAP。如果客户端已经知道服务端相关信息,客户端可以不访问中介端而直接访问服务端。这个中介端被称作 UDDI 服务器,其英文全称为 Universal Description, Discovery and Integration,可译为"通用描述、发现与集成服务"。Web Services 的服务端和客户端都必须知道这台 UDDI 服务器。

可以把客户端看成是买东西的顾客,把服务器端看成是卖东西的商家,把 UDDI 看成是市场。就以淘宝网为例,淘宝网是人所周知的场所,卖东西的人知道,买东西的人也知道。商家要先把商品价格和销后等信息(可看成是 WSDL 文件)发布到淘宝网(可看成是 UDDI),即服务器端将 WSDL 文件发布到 UDDI 中;然后顾客通过淘宝网发现和理解商家提供的商品价格等信息,即客户端访问 UDDI 服务器查询商家的 WSDL 文件和理解 WSDL 文件内容;最后顾客购买商家的商品和服务,即是客户端访问了服务器端服务。

8.3.4 SOAP 简述

SOAP(Simple Object Access Protocol)中文名为简单对象访问协议。它是一种轻量的、简单的和基于 XML 的协议,它被设计成在 Web 上交换结构化的和固化的信息。SOAP 可以和现存的许多因特网协议和格式结合使用,包括超文本传输协议(HTTP)、简单邮件传输协议(SMTP)和多用途网际邮件扩充协议(MIME)。它还支持从消息系统到远程过程调用(RPC)等大量的应用程序。

通过 SOAP 协议实现的 Web 服务,使全世界的程序员编写的类、函数可以集合起来,建成一个巨大的函数库,而且是语言无关的。这为软件事业描绘了一个灿烂的发展前景,只要网络连在一起,就可以实现代码级的逻辑共享,过去的跨进程、跨机器、跨网络的通信问题全部都解决了,而且 HTTP 协议是能够穿越防火墙的(其实是防火墙一般不封锁 HTTP 协议的 80 端口,否则就无法上互联网)。

8.3.5 SOAP 类库

PHP 5 内置有 SOAP 扩展类库,通过 SOAP 类库,开发人员可以很容易地实现和调用 Web Services。默认情况,由于 PHP 5 并不加载 SOAP 类库,所以开发人员并不能直接使用 SOAP 类库。要使用该类库,需要编辑 PHP 配置文件加载 SOAP 库文件。在 PHPnow 安装目录\php-5.2.14-Win32\php-apache2handler.ini 文件(在 PHP 的安装目录下,也可编辑 php.ini)里,加入下面一行代码:

```
extension=php_soap.dll
```

这行代码表示 PHP 5 支持加载 SOAP 类库,同时,还需要重新启动 Apache Web 服务器,可在控制面板的管理工具中的组件服务里重启 Apache 服务,重启之后即可使用 SOAP 类库。

8.3.6 SOAP 类

SOAP 类库提供有 SoapClient 类和 SoapServer 类。SoapClient 类的作用是调用 Web Services。SoapClient 类通常被应用在 Web Services 的客户端应用程序中。它有两种操作形式：WSDL 模式和 Non-WSDL 模式。在 WSDL 模式中，构造器可以使用 WSDL 文件名作为参数，并从 WSDL 中提取服务所使用的信息。non-WSDL 模式中使用参数来传递要使用的信息。SoapClient 类提供了许多调用 Web 服务的方法，其中，SoapClient::__soapCall()是最重要的，这个方法可以用来调用服务中的某个操作。

SoapServer 类可以用来提供 Web Services。与 SoapClient 类似，SoapServer 也有两种操作模式：WSDL 模式和 Non-WSDL 模式。这两种模式的意义与 SoapClient 的两种模式一样。在 WSDL 模式中，服务实现了 WSDL 提供的接口；在 Non-WSDL 模式中，参数被用来管理服务的行为。在 SoapServer 类的众多方法中，有 3 个方法比较重要，它们是 SoapServer::setClass、SoapServer::addFunction 和 SoapServer::handle。SoapServer::setClass 方法用来指定实现 Web Services 的类。被 SoapServer::setClass 所指定的类中的所有公共方法都将成为 Web Services 的操作(operation)。SoapServer::addFunction 方法用来添加一个或多个作为 Web Services 操作(operation)的函数。SoapServer:: handle 方法指示 Web Services 脚本开始处理进入的请求。Web Services 脚本是用 PHP 脚本写的一个或多个 SoapServer 对象的实例。尽管可以有不止一个的 SoapServer 对象，但通常的习惯是一个脚本只拥有一个 SoapServer 实例。在调用 SoapServer::handle 方法之前，Web Services 脚本会使用设置在 SoapServer 对象实例上的任何信息来处理进入的请求和输出的相应。

8.3.7 使用 SOAP 类实现和调用 Web 服务

如果 Web 服务位置不会变更，而且客户端知道 Web 服务的相关信息，那么就可以使用 Non-WSDL 模式实现和调用 Web 服务。WSDL 模式需要 WSDL 文件支持，而 WSDL 文件较复杂，通常需要借助工具自动生成 WSDL 文件，在此不深入介绍 WSDL 模式的实现，读者可参看相关资料。下面是 Non-WSDL 模式下实现和调用 Web 服务的范例。

【例 8-3-2】 (代码位置：\8\SOAP 目录)

服务端文件 ServerSoap.php 的内容如下：

```
<?php
class service
{
    public function HelloWorld()
    {
        return "HelloWorld";
    }
    public function Add($a,$b)
    {
```

```
        return $a+$b;
    }
}
$server=new SoapServer(null,array('uri'=>"abcd"));     //使用 Non-WSDL 模式
$server->setClass("service");
$server->handle();
?>
```

服务端先创建一个 SoapServer 对象,该对象支持 Non-WSDL 模式,然后设置处理 SOAP 请求的类为 service,类 service 中的两上公有方法成为 Web 服务,最后使用 handle 函数开始处理接收到的 SOAP 请求。

客户端文件 ClientSoap.php 的内容如下:

```
<?php
try{
    $soap=new SoapClient(null,array(
            "location"=>"http:          //127.0.0.1/ServerSoap.php",
            "uri"     =>"abcd",         //资源描述符服务器和客户端必须对应
            "style"   =>SOAP_RPC,
            "use"     =>SOAP_ENCODED
            ));                         //创建一个 SoapClient 实例
    echo $soap->Add(3,2);               //调用有参数的 Web 服务
    echo "<br/>";
    echo $soap->HelloWorld();           //调用无参数的 Web 服务
}catch(Exction $e){
    echo print_r($e->getMessage(),true);
}
?>
```

客户端先创建一个 SoapClient 的实例,指定要访问的 Web 服务的 URL 和 URI,它支持 Non-WSDL 模式,然后像普通对象一样调用 Web 服务中的方法。把上述两个文件部署到 http 根目录下,在浏览器中输入 http://127.0.0.1/ClientSoap.php,运行结果如下:

```
5
HelloWorld
```

第 9 章



PHP 超级全局变量

PHP 超级全局变量数组(Super Global Array),又称为 PHP 预定义数组,是由 PHP 引擎内置的,不需要开发者重新定义。在 PHP 脚本运行时,PHP 会自动将一些数据放在超级全局变量数组中,放在超级全局数组中的数据包括服务器端数据和客户端数据。应用可以很方便地利用这些数据实现特定的功能,比如读取客户端 IP 以防止相同 IP 重复投票。在 PHP 网站开发中,为了满足网站的需要,时常需要对 PHP 环境变量进行设置和应用,在虚拟主机环境下,有时更需要通过 PHP 超级全局变量数组来对 PHP 环境变量等值进行设置,为此有必要对 PHP 超级全局变量数组有所熟悉。PHP 超级全局变量数组主要有 $GLOBALS、$_SERVER、$_GET、$_POST、$_COOKIE、$_FILES、$_ENV、$_REQUEST 和 $_SESSION 等。$_GET 和 $_POST 主要是用来获取 HTML 表单提交的数据。$_COOKIE 和 $_SESSION 主要是用来存储和管理客户端浏览器和服务器端间的会话数据。$_FILES 主要是获取文件上传时提交的数据。$_REQUEST 主要是获取 HTML 表单中的数据,它和 $_GET 和 $_POST 相同,但不区分 GET 和 POST 方法,$_GET、$_POST、$_COOKIE 中的所有数据都可以通过 $_REQUEST 来获取,可以通过 print_r 函数分别输出 $_REQUEST 或者 $_COOKIE 等进行比较。$_SERVER 是一个包含服务器端相关信息的 PHP 全局环境变量。网站大多都会用到 $_GET、$_POST、$_REQUEST、$_SESSION 和 $_SERVER 等变量,下面着重介绍 $_SESSION 和 $_SERVER 变量及相关知识。

9.1 Session

9.1.1 Session 概念

现要求实现以下任务:在浏览器打开后和关闭前,记录浏览器访问网站的次数并显示出来。开发人员可能会写如下代码:

```
<?php
static $i=0;
$i++;
echo "浏览器刷新次数:$i <br/>";
?>
```

打开浏览器运行这段代码，无论单击浏览器中的“刷新”按钮多少次，会发现运行结果始终如下：

```
浏览器刷新次数：1
```

代码中已经实现静态变量 $i 递加，为什么刷新次数没有发生变化呢？因为服务器是无状态的，面对网络中的众多客户，它不能记住客户状态，也就是无法知道变量 $i 到底属于哪个客户，换句话说，客户端上一次的访问、这次的访问和下一次的访问，服务器是无法知道它们之间的联系的，它们到底是来自同一个客户还是来自不同的客户请求。如果不解决这个问题，那么很难实现类似电子商务的网站，因为服务器无法知道前后购买的商品是来自同一个用户还是另一个用户。下面是关于 Session 的范例，范例把前述代码改写成如下代码。

【例 9-1-1】 （代码位置：\9\testSession.php）

```
<?php
  session_start();
  if(isset($_SESSION["i"]))
  {
    $i=$_SESSION["i"];
  }
  else $i=0;
  $i++;
  $_SESSION["i"]=$i;
  echo "浏览器刷新次数：$i <br/>";
?>
```

打开浏览器运行这段代码，当刷新页面时，刷新次数会递增，而另打开一个浏览器运行这段代码进行刷新，会发现不同的浏览器对刷新次数互不影响。下面是同一个浏览器刷新 5 次后的运行结果。

```
浏览器刷新次数：5
```

在这段代码里用到了 Session 技术，从例中可知，通过 Session 能知道变量 $_SESSION["i"]是来自同一个用户，即能区别不同用户的请求，从而服务器能区别对待不同的用户。Session 在 Web 技术中占有非常重要的分量，通常将 Session 翻译成会话。那么什么是 Session？Web 中的 Session 指的是，当用户在浏览某个网站时，即从进入网站到浏览器关闭所经过的这段时间，将该用户的信息进行存储管理的一种容器。

为什么要用 Session 来存储管理用户的信息呢？由于 HTTP 协议是一种无状态的通信协议，无法得知用户的浏览状态，因此必须通过 Session 记录用户的有关信息，以供用户再次以此身份对 Web 服务器提供要求时作确认，例如，在某些网站中常常要求用户登录，但怎么知道用户已经登录了呢？如果没有 Session 的话，登录信息是无法保留的，那样用户在每一页网页中都要提供用户名和密码。当然，Session 不仅用于用户身份认证功能，还有其他方面的应用，它最主要的作用是用于用户身份认证、程序状态记录和页

面之间传递参数等。

9.1.2 Session 工作原理

浏览器和服务器采用 HTTP 无状态的方式进行通信,为了保持客户端的状态,使用 Session 来达到这个目的。然而服务端如何标识不同的客户端或用户呢? 在 Session 机制中,采用一个唯一的 session_id(默认名为 PHPSESSID)来标示不同的用户,这个 session_id 由服务器生成,分别保存在用户端和服务器端,浏览器每次向服务器发出请求都会带上由服务器为它生成的 session_id。

Session 工作机制如何运作? 当客户端访问服务器时,服务器根据需求设置 Session,将会话信息保存在服务器上,同时,生成标识 Session 的 session_id 并将 session_id 传递给客户端浏览器,浏览器将这个 session_id 保存在内存中(还有其他的存储方式,例如写在 URL 中),将此称之为无过期时间的 cookie。浏览器关闭后,这个 cookie 就清掉了,它不会存在用户的 cookie 临时文件中。以后浏览器每次请求都会额外加上这个参数值,服务器会根据这个 session_id 取得客户端的数据状态。如果客户端浏览器意外关闭,服务器保存的 Session 数据不是立即释放,此时数据还会存在,只要服务器知道该 session_id,就可以继续通过请求获得此 Session 的信息;虽然后台的 Session 通常不会立即释放,但是 Session 的保存有一个过期时间,一旦超过规定时间没有客户端请求时,它就会自动清除这个 Session。

9.1.3 Session 函数和$ _SESSION 变量

PHP 提供了 Session 相关函数用于管理变量 Session,本节介绍 Session 相关函数和读写 Session 变量的方法。

函数 session_start 用来开始一个会话或者返回已经存在的会话,格式如下:

```
boolean session_start(void)
```

说明:这个函数没有参数,且返回值均为 true。最好将这个函数置于最先,而且在它之前不能有任何输出,否则会报警。

在 PHP 中 Session 的写入和读取是通过预定义数组 $_SESSION 的调用和读取来完成的,Session 的写格式如下:

```
$ _SESSION["session_name"]=变量
```

参数 session_name 表示会话变量的名称。Session 的读格式如下:

```
变量=$_SESSION["session_name "]
```

函数 isset 用于检测 Session 是否已经被存在,格式如下:

```
bool isset($_SESSION['session_name'])
```

函数 unset 用于注销已经建立的 Session 变量,格式如下:

```
unset($_SESSION['session_name']))
```

函数 session_encode 将当前会话数据编码为一个字符串，格式如下：

```
string session_encode(void)
```

该函数将返回一个字符串，返回的字符串中包含全局变量中各变量的名称与值，形式如下：a|s:12:"it is a test";c|s:4:"lala"。其中，a 是变量名，s:12 代表变量 a 的值“it is a test”的长度是 12，变量间用分号“;”分隔。

函数 session_decode 从字符串解码 Session 资料，格式如下：

```
boolean session_decode(string data)
```

这个函数可将 Session 信息解码，该函数将返回布尔值，成功则返回逻辑值 true。

函数 session_destroy，结束当前的会话，格式如下：

```
boolean session destroy(void)
```

这个函数将结束当前的 Session，此函数没有参数，且返回值均为 true。

函数 session_unset 用来删除所有已注册的变量，格式如下：

```
void session_unset(void)
```

这个函数和 session_destroy 不同，它不结束会话。该函数返回布尔值。

9.1.4 读和写 Session 变量

下面是读写 Session 变量的范例。

【例 9-1-2】 （代码位置：\9\testSessionRW.php）

```
<?php
  session_start();          //函数 session_start 置于前面，其前面不要有 echo 等输出 HTML 内容
  $_SESSION["age"]=50;                   //写 Session 变量
  $_SESSION["name"]="李明";
  $_SESSION["月薪"]=3500.9;
  echo $_SESSION["age"];                 //读 Session 变量
  echo "<br/>";
  echo $_SESSION["name"];
  echo "<br/>";
  echo $_SESSION["月薪"];
?>
```

上例中先写 3 个 Session 变量，然后再读出这 3 个变量并打印出来，值得注意的是，session_start 必须放置在前面，其前面不能有任何输出语句。运行结果如下：

```
50
李明
3500.9
```

9.1.5 删除 Session 变量

下面是删除 Session 变量的范例。

【例 9-1-3】 (代码位置:\9\testSessionDel. php)

```
<?php
  session_start();
  $_SESSION["age"]=50;
  $_SESSION["name"]="李明";
  $_SESSION["月薪"]=3500.9;
  unset($_SESSION["name"]);        //删除变量$_SESSION["name"]
  echo $_SESSION["age"];
  echo "<br/>";
  echo $_SESSION["name"];
  echo "<br/>";
  echo $_SESSION["月薪"];
?>
```

上例由于删除了变量$_SESSION["name"],所以读该变量时会提示出错,运行结果如下:

```
50
Notice: Undefined index: name in C:\PHP\htdocs\test2_1.php on line 11
3500.9
```

9.1.6 检测 Session 变量

下面是检测 Session 变量的范例。

【例 9-1-4】 (代码位置:\9\testSessionDetect. php)

```
<?php
  session_start();
  $_SESSION["age"]=50;
  $_SESSION["name"]="李明";
  $_SESSION["月薪"]=3500.9;
  unset($_SESSION["name"]);
  echo $_SESSION["age"];
  echo "<br/>";
  if(isset($_SESSION["name"]))     //检测变量$_SESSION["name"]是否存在
      echo $_SESSION["name"];
  else echo "检测结果:已删除";
  echo "<br/>";
  echo $_SESSION["月薪"];
?>
```

例中通过函数 isset 检测变量 $_SESSION["name"]是否存在,由于已被删除,所以该变量不存在,运行结果如下:

```
50
检测结果:已删除
3500.9
```

9.1.7 Session 编解码

当 Session 中的数据较大而且不常用时,可以把 Session 中的数据编码成字符串,把这些字符串暂时存储到硬盘中,当需要的时候再恢复 Session 数据。下面是 Session 编码的范例。

【例 9-1-5】 (代码位置:\9\testSessionEncode.php)

```
<?php
  session_start();
  $_SESSION["age"]=50;
  $_SESSION["name"]="李明";
  $_SESSION["月薪"]=3500.9;
  $str=session_encode();
  echo $str;
  session_destroy();
?>
```

程序运行结果如下:

```
age|i:50;name|s:4:"李明";月薪|d:3500.90000000000009094947017729282379150390625;
```

下面是 Session 解码的范例。

```
<?php
  session_start();
  $str="爱好|s:10:\"唱歌和游戏\"";
  $str=session_decode($str);      //字符串解码生成 session 数据
  echo $_SESSION["爱好"];
  session_destroy();
?>
```

运行结果如下:

```
唱歌和游戏
```

9.1.8 销毁 Session 会话

每一个会话 PHP 都会用一个唯一的 session_id 进行标识,不同的会话其标识符是不同的,下面的范例将打印销毁会话前后的 session_id。

【例 9-1-6】 (代码位置:\9\testSessionDestory.php)

```
<?php
  session_start();
  $_SESSION["age"]=50;
  $id=session_id();                 //获取当前会话的 id
  echo "id: $id";                   //输出会话 id
  echo "<br/>";
  session_destroy();                //结束会话
  $id=session_id();                 //获取会话结束后的 id
  echo "id: $id";                   //输出会话 id
  echo "<br/>";
  echo $_SESSION["age"];
?>
```

由于销毁了会话,因此销毁前后的 session_id 是不同的,但销毁前后的变量 $_SESSION["age"]却依旧存在,不同的是,销毁前的变量 $_SESSION["age"]是会话变量,而销毁后的变量 $_SESSION["age"]则不是会话变量,只作为普通变量存在。程序运行结果如下:

```
id: rvn2iqlmnrnlc7oeovt74mehu3
id:
50
```

通常结束一个会话时,应该要先使用函数 session_unset 释放 Session 中的变量等资源,再使用函数 session_destroy 结束会话。下面是销毁会话资源的范例。

【例 9-1-7】 (代码位置:\9\testSessionVarDestory.php)

```
<?php
  session_start();
  $_SESSION["age"]=50;
  $id=session_id();
  echo "id: $id";
  echo "<br/>";
  session_unset();                  //释放 session 资源
  session_destroy();                //销毁会话
  $id=session_id();
  echo "id: $id";
  echo "<br/>";
  echo $_SESSION["age"];
?>
```

例中由于已释放资源和结束会话,因此 session_id 输出为空,读变量 $_SESSION["age"]时出错,运行结果如下:

```
id: 4pq6kdamtuslqm6ghncspsf1d1
```

```
id:
Notice: Undefined index
```

9.1.9 Session 时间

Session 数据通常在浏览器关闭后就会马上过期被释放。如果想让用户在浏览器关闭后的一段时间内(例如 5 分钟内)还可继续访问 Session 数据,或在规定的时间内用户没有访问 Session 才让 Session 数据过期,那么可以修改 PHP 配置文件,该文件是 PHPnow 安装目录\php-5.2.14-Win32\php-apache2handler.ini(或 php.ini)文件,在文件中找到下面一行代码:

```
session.cookie_lifetime=0
```

修改成如下代码并重启 Apache 服务器:

```
session.cookie_lifetime=300
```

9.2 Session 防页面刷新应用

用户提交表单时可能因为网速的原因或者网页被恶意刷新,致使同一条记录重复插入到数据库,同时,恶意刷新还会加大服务器负荷压力。要解决这个问题,可以在服务器端和客户端着手避免重复提交。在服务器端方面,可以利用 Session 解决这一问题。用 Session 解决表单重复提交工作的原理如下:①在客户访问表单时,PHP 生成随机数并把随机数和时间保存到 Session 和 HTML 隐藏元素中;②正常提交表单后将生成新的随机数和时间,将它们也保存到 Session 中覆盖原先的值;③每次刷新页面或访问页面时,将 HTML 隐藏元素和 Session 中的时间和随机数进行比较,匹配则允许访问页面,否则认为是在恶意刷新页面而拒绝访问。

下面是用 Session 变量实现防页面刷新的范例,由 myenter.php 和 mybusiness.php 两个文件组成。如果通过页面 myenter.php 访问页面 mybusiness.php,则允许访问页面 mybusiness.php 中的业务逻辑;如果通过刷新方式(单击浏览器刷新按钮)不断访问 mybusiness.php 则将被拒绝访问业务逻辑。

【例 9-2-1】 (代码位置:\9\Refresh 目录)

文件 myenter.php 内容如下:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>b.html</title>
<?php
session_start();
$code=mt_rand(0,1000000);
```

```
$_SESSION["code"]=$code;
?>
</head>
<body>
<form action="mybusiness.php" method="post">
<input type="hidden" name="code" value="<?=$code?>"/>
<input type="submit"/>
</form>
</body>
</html>
```

文件 mybusiness.php 内容如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>b.html</title>
</head>
<body>
<?php
  session_start();
  if(isset($_REQUEST["code"]))
  {
      if($_REQUEST["code"]==$_SESSION["code"])
      {  //业务逻辑
         $code=mt_rand(0,1000000);
         $_SESSION["code"]=$code;
         echo "正常处理";
      }
      else
      {
          echo "请不要刷新本页面或重复提交表单!";
      }
  }
  else echo "请从 myenter.php 访问本页面";
?>
</body>
</html>
```

如通过页面 myenter.php 访问页面 mybusiness.php，则允许访问业务逻辑并显示“正常处理”；如通过刷新方式访问 mybusiness.php，则将被拒绝访问业务逻辑并显示“请不要刷新本页面或重复提交表单!”；如不是通过 myenter.php 页面而是其他页面访问 mybusiness.php 页面，则会显示“请从 myenter.php 访问本页面”。

9.3 $ _SERVER数组变量

9.3.1 $ _SERVER变量参数

$_SERVER 是服务器超级全局变量数组，它可以在任何页面中使用。这个数组里面保存了很多类型的数据，例如服务器 IP、客户 IP 和浏览器型号等。这些信息对应用程序很有意义，可以利用客户 IP 信息限制客户重复投票。下面是 $_SERVER 的详细参数和说明。

$_SERVER['PHP_SELF']表示当前正在执行脚本的文件名，与 document root 相关。在 HTML 表单中，如执行文件是本身，可以在 ACTION 中使用 $_SERVER['PHP_SELF']，好处是当执行文件名有变动时可以不必频繁替换 ACTION 中的文件名。

$_SERVER['SERVER_NAME']表示当前运行的 PHP 程序所在服务器主机的名称。

$_SERVER['REQUEST_METHOD']表示访问页面时的请求方法，即 GET、HEAD、POST 和 PUT。

$_SERVER['DOCUMENT_ROOT']表示当前运行的 PHP 程序所在的文档根目录。

$_SERVER['HTTP_REFERER']表示链接到当前页面的前一页面的 URL 地址，这在页面跳转功能中非常有用。

$_SERVER['REMOTE_ADDR']表示正在浏览当前页面访问者的 IP 地址。

$_SERVER['REMOTE_HOST']表示正在浏览当前页面用户的主机名。

$_SERVER['REMOTE_PORT']表示正在浏览的用户连接到服务器时所使用的端口。

$_SERVER['SCRIPT_FILENAME']表示当前执行脚本的绝对路径名。

$_SERVER['SERVER_PORT']表示服务器所使用的端口。

$_SERVER['SCRIPT_NAME']表示包含当前脚本的路径，这在页面需要指向自己时非常有用。

$_SERVER['REQUEST_URI']表示访问此页面所需的 URI，如"/index.html"。

$_SERVER['PHP_AUTH_USER']应用在 HTTP 用户登录认证功能中，这个变量是用户输入的用户名。

$_SERVER['PHP_AUTH_PW']应用在 HTTP 用户登录认证功能中，这个变量是用户输入的密码。

$_SERVER['AUTH_TYPE']应用在 HTTP 用户登录认证功能中，这个变量是认证的类型。

注意：当在 PNPnow 安装目录\php-5.2.14-Win32\ php-apache2handler.ini 中的 register_globals 设置为 On 时，上述变量在所有 PHP 程序脚本中都可用，也就是 $_SERVER 数组被分离了。当然为了安全考虑，还是不要将 register_globals 打开为

好，其默认值是 Off。下面是关于 register_globals 设置全局变量的范例，由 testRegGlobalVar.htm 和 testRegGlobalVar.php 文件组成，testRegGlobalVar.htm 提交表单变量 user_name 给 testRegGlobalVar.php，testRegGlobalVar.php 提供了两种方式访问该表单变量，一种是 $_GET 全局变量，另一种是 $user_name 全局变量，如果 register_globals 值为 Off，那么全局变量 $user_name 将失效。

【例 9-3-1】（代码位置：\9\Register_globals 目录）

文件 testRegGlobalVar.htm 内容如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form name="frmTest" id="frmTest" action="testRegGlobalVar.php" >
<input type="text" name="user_name" id="user_name">
<input type="password" name="user_pass" id="user_pass">
<input type="submit" value="login">
</form>
</body>
</html>
```

文件 testRegGlobalVar.php 内容如下：

```
<?php
echo $_GET['user_name'];
echo $user_name;            //如 php-apache2handler.php 中的 register_globals=On,那么该语句有效
?>
```

9.3.2 打印$_SERVER数组变量

如果记不住 $_SERVER 变量的详细参数，可以通过下面的代码输出 $_SERVER 变量的所有信息。其他全局变量的信息也可以用这个方法打印出来。

【例 9-3-2】（代码位置：\9\testServerVar.php）

```
<?php
  foreach($_SERVER as $key=>$value)
  {
      echo '$_server['.$key.']='.$value;
      echo "</br>";
  }
?>
```

程序运行结果如下：

```
$_SERVER[HTTP_ACCEPT]= */*
$_SERVER[HTTP_ACCEPT_LANGUAGE]=zh-CN
$_SERVER[HTTP_USER_AGENT]=Mozilla/4.0(compatible;MSIE 7.0;Windows NT 6.1;Trident/5.0;
    SLCC2;.NET CLR 2.0.50727;.NET CLR 3.5.30729;.NET CLR 3.0.30729;Media Center PC 6.0;.NET4.
    0C;.NET4.0E)
$_SERVER[HTTP_ACCEPT_ENCODING]=gzip,deflate
$_SERVER[HTTP_HOST]=127.0.0.1:82
$_SERVER[HTTP_CONNECTION]=Keep-Alive
$_SERVER[PATH]=C:\windows\system32;C:\windows;
$_SERVER[SystemRoot]=C:\windows
$_SERVER[COMSPEC]=C:\windows\system32\cmd.exe
$_SERVER[PATHEXT]=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
$_SERVER[WINDIR]=C:\windows
$_SERVER[SERVER_SIGNATURE]=
$_SERVER[SERVER_SOFTWARE]=Apache/2.2.16(Win32)PHP/5.2.14
$_SERVER[SERVER_NAME]=127.0.0.1
$_SERVER[SERVER_ADDR]=127.0.0.1
$_SERVER[SERVER_PORT]=82
$_SERVER[REMOTE_ADDR]=127.0.0.1
$_SERVER[DOCUMENT_ROOT]=C:/PHP/htdocs
$_SERVER[SERVER_ADMIN]=admin@phpnow.org
$_SERVER[SCRIPT_FILENAME]=C:/PHP/htdocs/ testServerVar.php
$_SERVER[REMOTE_PORT]=7363
$_SERVER[GATEWAY_INTERFACE]=CGI/1.1
$_SERVER[SERVER_PROTOCOL]=HTTP/1.1
$_SERVER[REQUEST_METHOD]=GET
$_SERVER[QUERY_STRING]=
$_SERVER[REQUEST_URI]=/ testServerVar.php
$_SERVER[SCRIPT_NAME]=/ testServerVar.php
$_SERVER[PHP_SELF]=/ testServerVar.php
$_SERVER[REQUEST_TIME]=1362910089
```

9.3.3 $_SERVER数组的4个重要变量

把例9-3-2中的文件testServerVar.php更名为index.php并复制到PHP安装目录下的任意子目录中，比如mytest目录。接下来通过3个例子讲述数组$_SERVER中的QUERY_STRING、REQUEST_URI、SCRIPT_NAME和PHP_SELF等参数的作用和区别。如果直接在浏览器中打开主页http://127.0.0.1/mytest，则$_SERVER变量的结果如下：

```
$_SERVER[QUERY_STRING]=
$_SERVER[REQUEST_URI]=/mytest/
$_SERVER[SCRIPT_NAME]=/mytest/index.php
```

```
$_SERVER[PHP_SELF]=/mytest/index.php
```

在该例中，$_SERVER["QUERY_STRING"]的值是空，$_SERVER["REQUEST_URI"]的值是"/mytest"，$_SERVER["SCRIPT_NAME"]的值是"/mytest/index.php"，$_SERVER["PHP_SELF"]的值是"/mytest/index.php"。注意：此处的"="符号的并不是表示赋值符号，而是表明"="符号左边的变量的值是"="符号右边的值。如果直接在浏览器中打开 http://127.0.0.1/mytest/?p=222，则 $_SERVER 变量的结果如下：

```
$_SERVER[QUERY_STRING]=p=222
$_SERVER[REQUEST_URI]=/mytest/?p=222
$_SERVER[SCRIPT_NAME]=/mytest/index.php
$_SERVER[PHP_SELF]=/mytest/index.php
```

如果直接在浏览器中打开 http://127.0.0.1/mytest/index.php?p=222&q=biuuu，则 $_SERVER 变量的结果如下：

```
$_SERVER[QUERY_STRING]=p=222&q=biuuu
$_SERVER[REQUEST_URI]=/mytest/index.php?p=222&q=biuuu
$_SERVER[SCRIPT_NAME]=/mytest/index.php
$_SERVER[PHP_SELF]=/mytest/index.php
```

从上面的范例可知，变量 $_SERVER["QUERY_STRING"]用来获取查询语句，获取的是?后面的值；变量 $_SERVER["REQUEST_URI"] 用来获取 http://127.0.0.1 后面的值，包括符号"/"；变量 $_SERVER["SCRIPT_NAME"] 用来获取当前脚本的路径，比如，当前脚本的路径是"/mytest/index.php"；变量 $_SERVER["PHP_SELF"] 获取的是当前正在执行脚本的文件名。本节通过实例详解了数组 $_SERVER 中容易混淆的 QUERY_STRING、REQUEST_URI、SCRIPT_NAME 和 PHP_SELF 这 4 个变量之间的作用和区别。

第 10 章 chapter 10

Xajax 和 JSON 技术

10.1 什么是 AJAX

在学习 Xajax 之前需要先了解 AJAX。AJAX 是一个比较“时尚”的 Web 开发技术。AJAX 全称为 Asynchronous JavaScript and XML(异步 JavaScript 和 XML),是一种创建交互式网页应用的网页开发技术。AJAX 并不是一门语言,而是一种技术,但不是一项新的技术,它可以创建更好、更快且交互性更强的 Web 应用程序。从名称就可以知道,它是把 XML、JavaScript 和异步通信等多项技术按一定的方式组合在一起,在共同的协作中发挥各自的作用的技术。AJAX 使用 XHTML 和 CSS 标准实现 Web 页面,使用 DOM 实现动态显示和交互,使用 XML 进行数据交换与处理,使用 JavaScript 绑定和处理所有数据。

AJAX 最大的优点就是页面无须刷新就可以更新页面内容和数据,减少用户实际和心理上的等待时间,用户不再遇到因为页面刷新导致浏览器长时间空白,甚至停止响应的糟糕结果,这给了用户较佳的体验效果。AJAX 基于标准化的 XML,被广泛使用并且支持良好,利于维护和修改。AJAX 调用外部数据很方便,在需要页面与数据分离的情况下,可以应用 AJAX 获取数据。而且,使用异步方式与服务器通信,不需要打断用户的操作,具有更加迅速的响应能力。除此,AJAX 技术可以把以前一些服务器负担的工作转到客户端,利用客户端闲置的能力来处理响应,可以最大程度地减少冗余请求以及减轻服务器响应客户端带来的负担。

10.2 Xajax

10.2.1 什么是 Xajax

Xajax 是一个开源 PHP 类库,它能够整合 HTML、CSS、JavaScript 和 PHP,轻而易举地开发功能强大、基于 Web 的 AJAX 应用软件。使用 Xajax 开发的应用软件,无须重新调入页面,就能够异步调用服务器端的 PHP 函数和更新内容。在使用 Xajax 类库之前,需要到 Xajax 官方网站下载该类库,其官方网站是:http://www.xajaxproject.org/

en/download/。成书时Xajax最新版本是Xajax 0.6 beta 1,较正式的版本是Xajax 0.5,本书以Xajax 0.5版本介绍Xajax程序开发。

Xajax的主要目的是针对那些熟悉PHP编写但不大熟悉JS的开发者,可将JS透明化,直接在PHP代码里完成需要的操作,而不用去编写JS端的代码,比如说在用户完成注册后通过Xajax实现页面异步提交,服务检测通过与否,并返回简单的提示信息,这些代码都只需在PHP代码页里完成,而在前端页只需放置触发的地方和一个回显信息的容器之类。

10.2.2 Xajax工作原理

如果PHP应用软件要异步调用PHP函数,那么可借助Xajax的PHP对象来实现。PHP的Xajax对象已封装好了对应的JavaScript函数。当PHP应用程序异步调用PHP函数时,PHP的Xajax对象中的封装函数会使用JavaScript的XMLHttpRequest对象与服务器异步通信,服务器再调用对应的PHP函数,调用结束后,PHP函数返回值由Xajax生成一个XML响应传递给应用程序。XML响应包含了特定的指令和数据,它们可以被Xajax的JavaScript消息分析器解析,并且被用于更新应用程序的内容。

10.2.3 Xajax使用步骤

从Xaja官方网站下载xajax_0.5_standard.rar并解压到目录C:\PHP\htdocs,解压后文件目录结构如图10-2-1所示。

examples	2012/9/9 20:18	文件夹	
tests	2012/9/9 20:18	文件夹	
xajax_controls	2012/9/9 20:18	文件夹	
xajax_core	2012/9/9 20:18	文件夹	
xajax_js	2012/9/9 20:18	文件夹	
xajax_plugins	2012/9/9 20:18	文件夹	
copyright.inc	2009/1/25 15:59	PHP 文件	2 KB

图 10-2-1　Xajax 目录结构

在PHP文件中通过下面7个步骤可实现Xajax,具体步骤如下。

(1) 包含Xajax类库。

```
include 'xajax_core/xajax.inc.php';
```

(2) 实例化Xajax对象。

```
$ xajax=new xajax();
```

(3) 注册需要通过Xajax调用的PHP函数的名称。

```
$ xajax->registerFunction("myFunction");
```

(4) 编写注册的PHP函数,并且在函数之中使用XajaxResponse对象返回XML指令。

```
function myFunction($arg)
{
  $text=$arg;                          //$arg是客户传递过来的参数
  $objResponse=new xajaxResponse();    //实例化 xajaxResponse 对象
  //给 id 为 SomeElementId 的元素的 innerHTML 属性赋新值$text
  $objResponse->assign('SomeElementId','innerHTML',$text);
  return $objResponse;
}
```

(5) 在脚本输出任何信息之前,调用 Xajax 用来接管请求。

```
$xajax->processRequest();
```

(6) 在页面的<head>和</head>标签之间,告诉 Xajax 生成所需的 JavaScript。

```
$xajax->printJavascript();
```

(7) 在程序中,在 JavaScript 事件或者函数中调用前面注册的函数,其函数名需加上前缀 xajax_。

```
<div id="SomeElementId"></div>
<button onclick="xajax_myFunction('It worked!');">OK</button>
```

10.2.4 Xajax 实例

下面是一个完整的 Xajax 范例。

【例 10-2-1】 (代码位置:\10\testXajax.php)

```
<?php
include 'xajax_core/xajax.inc.php';
$xajax=new xajax();
$xajax->registerFunction("myFunction");

function myFunction($arg)
{
  $text=$arg;
  $objResponse=new xajaxResponse();
  $objResponse->assign('SomeElementId','innerHTML',$text);
  return $objResponse;
}
$xajax->processRequest();
echo '<?xml version="1.0" encoding="UTF-8"?>';
?>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
```

```
    <title>xajax example</title>
<?php
    //output the xajax javascript. This must be called between the head tags
    $xajax->printJavascript();
?>
</head>
<body>
<div id="SomeElementId"></div>
<button onclick="xajax_myFunction('It worked!');" >OK</button>
</body>
</html>
```

程序运行结果如图 10-2-2 所示，其中图 10-2-2(a)是程序首次运行的结果，图 10-2-2(b)是单击按钮后的结果。

OK　　　　It worked! OK

(a)　　　　(b)

图 10-2-2　Xajax 运行结果

10.3　Xajax 主要函数

Xajax 除了能让开发者异步调用 PHP 的函数外，还提供了许多功能强大的函数，通过这些函数可以实现对话框、修改 HTML 内容等功能。

1. 函数 Alert

函数形式：

```
function Alert($sMsg)
```

功能：使用一个消息框提醒用户。参数 $ sMsg 表示消息框中的消息。

实例：

```
$objResponse->Alert("Here is an alert.")
```

2. 函数 Assign

函数形式：

```
function Assign($sTarget,$sAttribute,$sData)
```

功能：将 $ sTarget 对象的 $ sAttribute 赋值为 $ sData。

实例：

```
$objResponse->Assign("myInput1","value",$DataFromDatabase)
```

3. 函数 append

函数形式：

```
function append($starget,$sattribute,$sdata)
```

功能：将 $ sData 添加于 $ sTarget 对象的 $ sAttribute 后。
实例：

```
$objResponse->Append("myDiv1","innerHTML",$DataFromDatabase2)
```

4. 函数 Prepend

函数形式：

```
function Prepend($sTarget,$sAttribute,$sData)
```

功能：将 $ sData 添加于 $ sTarget 对象的 $ sAttribute 前。
实例：

```
$objResponse->Prepend("myDiv2","innerHTML",$DataFromDatabase3)
```

5. 函数 Replace

函数形式：

```
function Replace($sTarget,$sAttribute,$sSearch,$sData)
```

功能：将 $ sTarget 对象的 $ sAttribute 中的 $ sSearch 替换为 $ sData。
实例：

```
$objResponse->Replace("myDiv3","innerHTML","xajax","<strong>xajax</strong>")
```

6. 函数 Script

函数形式：

```
function Script($sJS)
```

功能：调用一个 JavaScript 语句块。
实例：

```
$objResponse->Script("var x=prompt("Enter Your Name");")
```

7. 函数 ConfirmCommands

函数形式：

```
function ConfirmCommands($iCmdNumber,$sMessage)
```

功能：$ sMessage 作为提示消息让用户作出“是/否”选择，如果用户选择“是”，则接着该语句后的语句执行；否则，跳到第 $ iCmdNumber 条语句执行，如 $ iCmdNumber 值为 2 则表示程序会跳转到其后的第二条语句执行。

ConfirmCommands 函数实例如下。程序运行后会跳出对话框 1，当用户单击“是”按钮时会显示对话框 2 和对话框 4；当用户单击“否”按钮时会显示对话框 3 和对话框 4。

```
$objResponse->ConfirmCommands(2,"Are you sure you want to show two(2)more alerts?");  //对话框 1
$objResponse->Alert("This will only happen if the user presses OK.");                 //对话框 2
$objResponse->Alert("This also will only happen if the user presses Cancel.");        //对话框 3
$objResponse->Alert("This will always happen.");                                      //对话框 4
```

8. 函数 CreateInput

函数形式：

```
function CreateInput($sParent,$sType,$sName,$sId)
```

功能：在$sParent中添加一个类型为$sType、名为$sName、id为$sId的输入框。

实例：

```
$objResponse->CreateInput("testForm1","text","username","uesrname")
```

9. 函数 InsertInput

函数形式：

```
function InsertInput($sBefore,$sType,$sName,$sId)
```

功能：同上，区别在于在名为$sBefore的对象前添加。

10. 函数 InsertInputAfter

函数形式：

```
function InsertInputAfter($sAfter,$sType,$sName,$sId)
```

功能：同上，区别在于在名为$sAfter的对象后添加。

11. 函数 Remove

函数形式：

```
function Remove($sTarget)
```

功能：删除$sTarget对象。

12. 函数 Event

函数形式：

```
function Event($sTarget,$sEvent,$sScript)
```

功能：为$sTarget对象的$sEvent事件添加$sScript代码。

实例：

```
$objResponse->Event("divTest","onClick","alert('Something');")
```

13. 函数 Handler

函数形式：

```
function Handler($sTarget,$sEvent,$sHandler)
```

功能：将$sTarget对象的$sEvent事件处理程序与JavaScript函数$sHandler关联，一个事件可以有多个Handler。

实例：

```
$objResponse->Handler("Div","click","clickHandler1")
```

14. 函数 RemoveHandler

函数形式：

```
function RemoveHandler($sTarget,$sEvent,$sHandler)
```

功能：删除$sTarget对象的$sEvent事件的$sHandler关联。

实例：

```
$objResponse->Handler("Div","click","clickHandler1")
```

15. 函数 ScriptCall

函数形式：

```
function ScriptCall()
```

功能：调用一个JavaScript函数，第一个参数为JavaScript函数名，其余参数为JavaScript函数的参数。

实例：

```
$response->ScriptCall("myJSFunction","arg1",9432.12,array("myKey"=>"some value","key2"
=>$value2))
```

10.4 Xajax应用

10.4.1 更换主题颜色

很多网站提供选择主题的功能，主题功能让用户可根据不同的节日或喜好设置页面的颜色。下面提供的简易更换主题范例能让用户动态地更换文字颜色。

【例10-4-1】（代码位置：\10\testXajaxTheme.php）

```
<?php
  require('xajax_core/xajax.inc.php');
```

```
    $xajax=new xajax();
    //$xajax->configure('debug',true);
    $xajax->configure('javascript URI','../');
    function setColor($sColor)
    {
        $text='主题颜色';
        $objResponse=new xajaxResponse();
        $objResponse->assign('div1','style.color',$sColor);        //更换颜色
        $objResponse->assign('div1','innerHTML',$text);
        return $objResponse;
    }
    $reqSetColor=& $xajax->registerFunction('setColor');
    $reqSetColor->setParameter(0,XAJAX_INPUT_VALUE,'colorselect');
    $xajax->processRequest();
    echo '<?xml version="1.0" encoding="UTF-8"?>';
?>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>xajax example</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
<?php
    //output the Xajax JavaScript. This must be called between the head tags
    $xajax->printJavascript();
?>

</head>
<body style="text-align:center;">
    <div id="div1"> </div>
    <br/>
    请选择主题:<select id="colorselect" name="colorselect"
        onchange='<?php $reqSetColor->printScript();?>;'>
        <option value="black" selected="selected">Black</option>
        <option value="red">Red</option>
        <option value="green">Green</option>
        <option value="blue">Blue</option>
    </select>
</body>
</html>
```

程序运行结果如图 10-4-1 所示。单击下拉列表并选择 Red 选项,浏览器将显示红色的"主题颜色",如图 10-4-2 所示。



图 10-4-1 更换主题颜色界面



图 10-4-2 红色主题

10.4.2 乘法计算器

Xajax类库提供了一些实例，可以在Xajax解压目录下的examples目录中找到这些实例，目录multiply包含了计算器实例的3个文件：multiply. common. php、multiply. php和multiply. server. php。文件multiply. common . php主要用于显示计算器界面，multiply. server. php文件主要是实现乘法功能和更新计算结果，文件multiply. php是前面两者的纽带，这样的设计实现了界面与业务逻辑相分离，是较常见的设计思想之一。这3个文件的代码如下。

【例10-4-2】 （代码位置：\10\Calc目录）

文件multiply. php内容如下：

```
<?php
require("multiply.common.php");
echo '<?xml version="1.0" encoding="UTF-8"?>'
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <title>xajax Multiplier</title>
        <?php $xajax->printJavascript();?>
    </head>
    <body>
        <input type="text" name="x" id="x" value="2" size="3"/> *
        <input type="text" name="y" id="y" value="3" size="3"/> =
        <input type="text" name="z" id="z" value="" size="3"/>
        <input type="button" value="Calculate" onclick="xajax_multiply(document.
getElementById('x').value,document.getElementById('y').value);return false;"/
>
    </body>
</html>
```

文件multiply. common. php内容如下：

```
<?php
  require_once("xajax_core/xajax.inc.php");
  $xajax=new xajax("multiply.server.php");
  $xajax->configure('javascript URI','../../');
  $xajax->register(XAJAX_FUNCTION,"multiply");
?>
```

文件multiply. server. php内容如下：

```
<?php
```

```
    function multiply($x,$y)
    {
      $objResponse=new xajaxResponse();
      $objResponse->assign("z","value",$x * $y);
      return $objResponse;
    }
    require("multiply.common.php");
    $xajax->processRequest();
?>
```

在浏览器中输入 http://127.0.0.1/examples/multiply/multiply.php，程序运行结果如图 10-4-3 所示。

先输入相乘的两个数 2 和 3，再单击 Calculate 按钮，结果如图 10-4-4 所示。



图 10-4-3　乘法计算器　　**图 10-4-4　相乘结果**

10.4.3　Xajax 检测注册用户是否存在

很多系统在注册的时候都需要检查用户名是否已经被注册了，以往最简单的办法就是做个检查的按钮或链接，打开一个小窗口，提交信息到服务器，查询数据库得到结果后在小窗口显示。而使用 Xajax 的话，既不需要刷新页面也不需要开新窗口，直接在当前页面就可以检查和显示。下面来尝试这个功能，为了测试该功能，假设用户 admin 已存在。在实际应用中，只要把判断用户 admin 相关部分修改成数据库查询方式即可。其实现代码如下。

【例 10-4-3】（代码位置：\10\testXajaxCheckUser.php）

```
<?php
  include 'xajax_core/xajax.inc.php';                    //包含 Xajax 类
  //function check username
  function checkusername($testvalue)
  {
    if($testvalue=="admin")
        $checkresult='该用户已经注册';
    else $checkresult='可以注册';
    $objreponse=new xajaxResponse();
    $objreponse->assign('result','innerHTML',$checkresult);    //返回结果
    //return $objreponse->getXML();
    return $objreponse;
  }
  //设置 Xajax
  $xajax=new xajax();
  $xajax->registerFunction('checkusername');
```

```
    $xajax->processRequest();
    echo '<?xml version="1.0" encoding="UTF-8"?>';
?>
<html>
<head>
<title>Check Username</title>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<?php $xajax->printJavascript();?>
</head>
</head>
<body>
<form name="check">
    用户名:<input name="username" onblur="xajax_checkusername(this.value);"/>
    <font color="red"><div id="result"></div></font>
</form>
</body>
</html>
```

运行该例脚本,首先会看到一个用户名的输入框,如果输入 admin,然后在其他地方单击鼠标,那么输入框下面就会显示“该用户已经注册”的字样;如果输入 abcde,那么就会显示“可以注册”的字样。

10.4.4 Xajax结合表单进行登录

在此用 Xajax 来实现登录程序,这个功能是非常有用的,现在很多网站在首页上提供登录框,但由于首页通常显示内容比较多,如果通过提交表单来进行登录,一般会引起页面刷新,对网络传输造成不必要的负担。现在使用 Xajax 就可以实现不刷新页面的提交登录了。下面是 Xajax 在登录应用中的范例,范例中的合法用户名为 admin,合法密码为 123456。

【例 10-4-4】 (代码位置:\10\testXajaxLogin.php)

```
<?php
include 'xajax_core/xajax.inc.php';
function processForm($aFormValues)
{
    $objResponse=new xajaxResponse();
    $bError=false;
    $logstatus=false;
    $sForm="";

    if(trim($aFormValues['userid'])=="")
    {
        $objResponse->Alert("请输入账号");
        $bError=true;
```

```
}
if(trim($aFormValues['passwd'])=="")
{
    $objResponse->Alert("请输入密码");
    $bError=true;
}

if(!$bError)
{
  if((trim($aFormValues['userid'])!="admin")||( md5(trim($aFormValues ['passwd']))!
  =md5("123456")))
  {
      $objResponse->Alert("账号或密码错误,登录失败");
  }
  else
  {
      $logstatus=true;
  }
}
if($logstatus==true)
{
  $sForm="<table width='360' height='120' border='0' cellspacing='0' cellpadding=
  '2' bgcolor='DEEFF7'>";
  $sForm.="  <tr>";
  $sForm.="    <td width='35%' height='30'>  欢迎您</td>";
  $sForm.="    <td width='65%'>";
  $sForm.=      trim($aFormValues['userid']);
  $sForm.="    </td>";
  $sForm.="  </tr>";
  $sForm.="  <tr align='center'>";
  $sForm.="    <td height='30' colspan='2' colspan=2></td>";
  $sForm.="  </tr>";
  $sForm.="</table>";

}
else
{
  $sForm="<table width='360' border='0' cellspacing='0' cellpadding='2'>";
  $sForm.="  <form id='signupForm' action='javascript:void(null);'
             onsubmit='submitSignup();'>";
  $sForm.="    <tr>";
  $sForm.="      <td width='35%' height='30' align='center'>用户名：</td>";
  $sForm.="      <td width='65%'><input name='userid' type='text' class=
                 'form_input' size='18'/></td>";
```

```
        $sForm.="    </tr>";
        $sForm.="    <tr>";
        $sForm.="      <td height='30' align='center'>密 码：</td>";
        $sForm.="      <td><input name='passwd' type='password' class='form_input' size
                       ='18'/></td>";
        $sForm.="    </tr>";
        $sForm.="    <tr align='center'>";
        $sForm.="      <td colspan='2' align='center'><input type='submit' name='
                       loginsubmit' value='登录'/>";
        $sForm.="    </td>";
        $sForm.="    </tr>";
        $sForm.="  </form>";
        $sForm.="</table>";

    }
    $objResponse->assign("formDiv","innerHTML",$sForm);
    return $objResponse;
}
$xajax=new xajax();
$xajax->registerFunction("processForm");
$xajax->processRequest();
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>使用 Xajax 的登录脚本</title>
<?php $xajax->printJavascript();?>
</head>
<script>
function submitSignup()
{
    xajax_processForm(xajax.getFormValues("signupForm"));
    return false;
}
</script>
<body>
    <div id="formDiv">
        <form id="signupForm" action="javascript:void(null);" onsubmit=
        "submitSignup();">
            <table width="360" border="0" cellspacing="0" cellpadding="2">
                <tr>
                    <td width="35%" height="30" align="center">用户名：</td>
                    <td width="65%"><input name="userid" type="text" class=
                    "form_input" size="18"/></td>
```

```
            </tr>
            <tr>
                <td height="30" align="center">密 码:</td>
                <td><input name="passwd" type="password" class="form_input"
                size="18"/></td>
            </tr>
            <tr align="center">
                <td height="30" colspan="2" align="center"><input type=
                'submit' name='loginsubmit' value="登录"/>
                </td>
            </tr>
        </table>
    </form>
</div>
</body></html>
```

现在运行脚本,首先会看到一个登录界面,如果输入的用户名和密码分别为 admin 和 123456,就会在页面不刷新的情况进入登录后界面。

10.5 JSON

10.5.1 什么是 JSON

尽管有许多宣传关于 XML 如何拥有跨平台、跨语言的优势,然而,除非应用于 Web Services,否则,在普通的 Web 应用中,开发者经常为 XML 的解析伤透了脑筋,无论是服务器端生成或处理 XML,还是客户端用 JavaScript 解析 XML,都常常导致复杂的代码和极低的开发效率。实际上,对于大多数 Web 应用来说,根本不需要复杂的 XML 来传输数据,XML 的扩展性失去优势,许多 AJAX 应用甚至直接返回 HTML 片段来构建动态 Web 页面。和返回 XML 并解析它相比,返回 HTML 片段大大降低了系统的复杂性,但同时缺少了一定的灵活性。现在,JSON 为 Web 应用开发者提供了另一种数据交换格式。JSON 用于服务器端和客户端的数据传递,解决了 XML 的过多冗余标签造成数据传输量过大的问题,以及解决了服务器和客户端解析 XML 数据格式过于复杂等问题。JSON 与 XML 最大的不同在于 XML 是一个完整的标记语言,而 JSON 不是。这使得 XML 在程序判读上需要比较多的功夫。主要的原因在于 XML 的设计理念与 JSON 不同。XML 利用标记语言的特性提供了绝佳的延展性(如 XPath),在数据存储、扩展及高级检索方面具备对 JSON 的优势,而 JSON 则由于比 XML 更加小巧,以及浏览器的内建快速解析支持,使得其更适用于网络数据传输领域。

JSON 一种轻量级的数据格式,由资深的软件工程师 Douglas Crockford 开发。JSON 全称为 Javascript Object Notation,其中文翻译为 JavaScript 对象表示。和 XML 一样,JSON 采用完全独立于语言的文本格式,但是也使用了类似于 C 语言家族的习惯

(包括C、C++、C#、Java、JavaScript、Perl和Python等)，这使JSON成为理想的数据交换格式。由于JSON天生是为JavaScript准备的，因此，JSON的数据格式非常简单，可以用JSON传输一个简单的String、Number和Boolean型数据，也可以传输一个数组，或者一个复杂的Object对象。JSON用来描述数据结构，有名称/值、对象和数组等多种形式。

(1) 名称/值(key-value)形式。名称和值之间使用":"隔开，一般的形式如下：

```
{name:value}
```

一个名称是一个字符串；一个值可以是数字(整数或浮点数)、字符串(在双引号中)、逻辑值(true或false)、数组(在方括号中)、对象(在花括号中)和null。下面给出JSON名称/值实例：

```
"姓名": "李明"
```

(2) 对象(object)形式。一个对象以"{"开始，并以"}"结束。一个对象包含一系列非排序的名称/值对，每个名称/值对之间使用","分区。例如，一个Address对象包含如下名称/值对：

```
city:Beijing
street:Chaoyang Road
postcode:100025(整数)
```

用JSON表示如下：

```
{"city":"Beijing","street":" Chaoyang Road ","postcode":100025}
```

再如，用JSON表示人的一个对象：

```
{
"姓名" : "李明",
"年龄" : 24
}
```

(3) 数组(array)形式。JSON可以表示一个数组对象，使用[]包含所有元素，每个元素用逗号分隔，元素可以是任意的值。例如，以下数组包含了一个String、Number和Boolean型数据和一个null。

```
["abc",12345,false,null]
```

经过对象和数组2种结构就可以组合成复杂的数据结构了。下面给出复合数据结构范例。

```
{
    "姓名": "李明",
    "性别": true,
    "年龄": 25,
    "地址":
```

```
        {
            "街道": "北京路 21 号",
            "城市": "广州",
            "邮编": "111111"
        },
        "电话":
        [
            {
                "类型": "家庭固话",
                "电话号码": "027- 99999999"
            },
            {
                "类型": "传真",
                "电话号码": "027- 99999998"
            }
        ]
    }
```

例中是一个人对象数据结构，该对象包含“姓名”、“性别”、“年龄”、“地址”和“电话”等属性，其中，“地址”是对象数据结构，“地址”对象包含“街道”、“城市”和“邮编”属性，“电话”是数组结构，该数组包含有两个对象，每个对象包含“类型”和“电话号码”。

10.5.2 PHP 操作 JSON

从 PHP 5.2 版本开始，PHP 提供了内置函数 json_encode()和 json_decode()，这使 PHP 可以使用 JSON 与其他开发语言很好地传递数据。

```
mixed json_decode(string $json [,bool $assoc])
```

功能：将 JSON 数据转换为 PHP 变量。参数 json 表示待解码的 JSON 文本数据；参数 assoc 为 true 值时，返回联合数组，为 false 时返回的是对象，该参数可选，默认值是 false。该函数要求的字符串比较严格，具体要求如下：

(1) 使用 UTF-8 编码。

(2) 不能在最后元素有逗号。

(3) 不能使用单引号。

(4) 不能有\r 和\t，如果有，需要替换。

需要指出的是，在非 UTF-8 编码下，中文字符将不能被转换，结果会出来空值，所以，如果使用 gb2312 编写 PHP 代码，那么就需要将包含中文的内容使用 iconv 或者 mb 转为 UTF-8 再进行 json_encode。

```
string json_encode(mixed $value)
```

功能：该函数返回 value 值的 JSON 表达式。参数 value 要被编码的值，可以为除 resource 外的任何类型的参数，这个函数仅在 UTF-8 编码格式时起作用。当成功时则返

回编码后的JSON文本。

下面是函数json_encode的范例。

【例10-5-1】 (代码位置:\10\testJsonEncode.php)

```
<?php
$arr=array(
    'name'=>'Tom',
    'nick'=>'Cat',
    'contact'=>array(
        'email'=>'xx@qq.com',
        'website'=>'www.xx.com',
    )
);
$json_string=json_encode($arr);
echo $json_string;
?>
```

例中把PHP数组数据转换成JSON格式数据,数组$arr包含3个元素:name、nick和contact。name和nick是字符串类型数据,contact是一个包含有两个元素的数组。运行转换结果如下:

```
{"name":"Tom","nick":"Cat","contact":{"email":"xx@qq.com","website":"www.xx.com"}}
```

下面是把中文转换在JSON数据的范例。

【例10-5-2】 (代码位置:\10\testXajaxChineseToJson.php)

```
<?php
  $name='汤姆';
  $nick='小猫';
  $arr=array(
      'name'=>$name,
      'nick'=>$nick,
      'contact'=>array(
          'email'=>'xx@qq.com',
          'website'=>'www.xx.com',
      )
  );
  $json_string=json_encode($arr);
  echo $json_string;
?>
```

程序运行结果如下:

```
{"name":"\u6c64\u59c6","nick":"\u5c0f\u732b","contact":{"email":"xx@qq.com","website":"
www.xx.com"}}
```

由于函数json_encode并不支持非UTF-8格式的数据,GB 2312编码的中文并不能

成功转换，那是不是就无法把中文转换成 JSON 数据呢？借助函数 urlencode 和 urldecode 实现支持中文的 JSON，函数 urlencode 是把原字符串转换为 UTF-8 格式编码，这就满足了 json_encode 支持 UTF-8 的需求，而函数 urldecode 是把 UTF-8 编码转换成当前字符编码。在进行 json_encode 前先把包含有中文的字符串使用 urlencode 转换成 UTF-8 格式的字符串，然后执行 json_encode 把数据转换成 JSON 格式的数据，最后在显示 JSON 格式数据前接着使用函数 urldecode 把 JSON 数据解码为原编码。下面是支持中文的 JSON 范例。

【例 10-5-3】 (代码位置：\10\testXajaxJsonChinese. php)

```
<?php
  $name=urlencode('汤姆');
  $nick=urlencode('小猫');
  $arr=array(
      'name'=>$name,
      'nick'=>$nick,
      'contact'=>array(
          'email'=>'xx@qq.com',
          'website'=>'www.xx.com',
        )
  );
  $json_string=json_encode($arr);
  $json_string=urldecode($json_string);
  echo $json_string;
?>
```

程序运行结果如下：

```
{"name":"汤姆","nick":"小猫","contact":{"email":"xx@qq.com","website":"www.xx.com"}}
```

下面是 json_decode 的使用范例，例中把 JSON 数据转换成关联数组。

【例 10-5-4】 (代码位置：\10\testXajaxJsonDecode. php)

```
<?php
$jsonstr='
{
    "姓名": "李明",
    "性别": true,
    "年龄": 25,
    "地址":
    {
        "街道": "北京路 21 号",
        "城市": "广州",
        "邮编": "111111"
    },
```

```
        "电话":
        [
            {
                "类型": "家庭固话",
                "电话号码": "027-99999999"
            },
            {
                "类型": "传真",
                "电话号码": "027-99999998"
            }
        ]
}
';
$arr=json_decode($jsonstr,true);
print_r($arr);
?>
```

程序运行结果如下：

```
Array([姓名]=>李明 [性别]=>1 [年龄]=>25 [地址]=>Array([街道]=>北京路21号 [城市]=>广州 [邮编]=>111111)[电话]=>Array([0]=>Array([类型]=>家庭固话 [电话号码]=>027-99999999)[1]=>Array([类型]=>传真 [电话号码]=>027-99999998)))
```

10.5.3 JavaScript 处理 JSON

JSON 是 JavaScript 原生格式，这意味着在 JavaScript 中处理 JSON 数据不需要任何特殊的 API 或工具包。下面是 JavaScript 操作 JSON 对象的范例。

【例 10-5-5】 （代码位置：\10\testXajaxJavaScript.php）

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <script type="text/javascript">
        function testJSON()
        {
            var json={name:'Tom',sex:true,age:23};            //定义 JSON 对象
            var str="";
            //通过循环把对象中的属性读出来
            for(var item in json){
                str=str +"</br>" +item+"  "+typeof(item)+" " +eval("json."+item);
            }
            var mydiv=document.getElementById("mydiv");
            mydiv.innerHTML=str;
        }
```

```
    </script>
</head>
<body>
<button onclick="testJSON();">testJSON</button>
<div id="mydiv"></div>
</body>
</html>
```

程序运行后单击 testJSON 按钮，显示结果如图 10-5-1 所示。

```
testJSON
name string Tom
sex string true
age string 23
```

图 10-5-1 JavaScript 处理 JSON 对象

在 JavaScript 中如何把 JSON 字符串转换成 JSON 对象呢？利用 JavaScript 中的函数 eval 可以解决这个问题。eval 函数会评估一个给定的含有 JavaScript 代码的字符串，并且试图去执行包含在字符串里的表达式或者一系列的合法的 JavaScript 语句。eval 函数将把最后一个表达式或者语句所包含的值或引用作为返回值。在转化的时候需要将 JSON 字符串的外面加上一对圆括号，下面是 JSON 字符串转换成 JSON 对象的范例。

```
var jsonObject=eval("(" +jsonFormat +")");
```

为什么要加括号，加上圆括号的目的是迫使 eval 函数在评估 JavaScript 代码时强制将括号内的表达式(expression)转化为对象，而不是作为语句(statement)来执行。例如对象字面量{}，若不加外层的括号，那么 eval 会将大括号识别为 JavaScript 代码块的开始和结束标记，这样{}将会被认为是执行了一句空语句。所以下面两个执行结果是不同的：

```
alert(eval("{}");                    //返回值是 undefined
alert(eval("({})");                  //返回 object[Object]
```

JSON 格式的名字部分为什么要加引号？因为 eval 函数会将{foo:"bar"}解释成合法的 JavaScript 语句，而非表达式。但是人们往往想要的是让 eval 将这段代码解释成一个对象。所以 JSON 格式会强制在名字的外侧加上引号再结合圆括号，这样 eval 函数就不会错误地将 JSON 解释成代码块。下面举例说明 eval 函数错误解析语义。

```
alert(eval('{foo:"bar"}'));          //返回 bar,不能正确解析 JSON 对象
alert(eval('({"foo": "bar"})'));     //返回 JSON 对象,正确
```

下面是 JavaScript 把 JSON 字符串转换成 JSON 对象的范例。

【例 10-5-6】 (代码位置：\10\testXajaxJavaScriptConvert.php)

```
<html>
  <head>
  <script>
  function jsontest()
  {
    //JSON 字符串
```

```
    jsontext='{"name":"tom","sex":"false","age":"33"}';
    //解析 JSON 字符串并转换成 JSON 对象
    var aman=eval("("+jsontext+")");
    //取出 JSON 对象中的信息
    var str="";
    for(var item in aman){
      str=str +"</br>" +item +"  "+typeof(item)+" " +eval("aman."+item);
    }
    var mydiv=document.getElementById("mydiv");
    mydiv.innerHTML=str;
  }
  </script>
  </head>
  <body >
    <button onclick="jsontest();">testJSON</button>
    <div id="mydiv"></div>
  </body>
</html>
```

程序运行后单击 testJSON 按钮，显示结果如 10-5-2 所示。

```
testJSON
name string tom
sex string false
age string 33
```

图 10-5-2　JSON 转换结果

第 11 章 chapter 11

PHP 多媒体处理技术

11.1 图形图像处理技术概述

11.1.1 GD2 扩展库

PHP 中常用的图形图像处理类库有 GD、ImageMagick 和 GraphicsMagick 3 种类库。GD 库是 PHP 处理图形图像的扩展库,GD 库提供了一系列用来处理图形图像的 API,使用这些 API 可以动态地处理图片或者生成图片。PHP 要使用 GD 库,需要设置 PHPnow 安装目录\php-5.2.14-Win32 目录下的文件 php-apache2handler.ini,把文件中的 extension=php_gd2.dll 左边的分号“;”去掉,再重新启动 Apache Web 服务器即可。GD 库是由 C 语言编写的,GD 库支持 Perl、PHP 和其他语言开发。GD 库能处理 PNG、JPEG、GIF、WebP、XPM、BMP 和其他格式图片。在网站上 GD 库通常用来生成缩略图,或者用来对图片加水印,或者用来生成汉字验证码,或者对网站数据生成报表等。尽管 GD 库不局限于 Web 开发,但关于 GD 库的应用更多还是集中在网站开发上。GD 库最初由 Thomas Boutell 开发,但现在由 Pierre Joye 维护。GD 1.0 开始时是支持 GIF 图像的,但由于 GIF 使用了有版权争议的 LZW 算法,会引起法律问题,于是从 GD 1.6 开始,GD 库不再支持 GIF,而改为支持更好的、无版权争议的 PNG。直到 2004 年,GD 2.0.28 又恢复支持 GIF 图像,同时,它还支持 GIF 动画。PHP 5 安装附带了 GD 2.x 库,较早版本的 PHP 可能包含 GD 1.x 库,在较早版本的 PHP 中 GD 是一个额外的安装程序,因此也可能未安装 GD。

11.1.2 ImageMagick

ImageMagick 是一套功能强大、稳定而且免费的工具集和开发库,可以用来创建、编辑、组合和变换图片文件,能处理超过 100 种基本格式的图片文件,包括流行的 TIFF、JPEG、GIF、PNG、PDF 和 PhotoCD 等格式。ImageMagick 是一种免费使用的软件,开发人员可以利用它的源代码进行修改和发布。成书时 ImageMagick 的最新版本是 6.8.8-2,它的官方网站是 http://www.imagemagick.org。利用 ImageMagick,可以根据 Web 应用程序的需要动态生成图片,还可以对一张或一组图片进行改变大小、旋转、锐化、减色和

增加特效等操作,并将操作的结果以相同格式或其他格式进行保存。对图片的操作,既可以通过命令行进行,也可以用 C/C++、Perl、Java、PHP、Python 和 Ruby 编程语言来完成。同时 ImageMagick 提供了一个高质量的 2D 工具包,部分支持 SVG。现在,ImageMagick 的主要精力集中在性能、减少 bug 以及提供稳定的 API 和 ABI 上。ImageMagick 功能和特性如下。

(1) 格式转换:从一种图像格式转换成另一种图像格式,例如从 PNG 转 JPEG。

(2) 变换:缩放、旋转、裁剪、翻转和修剪图像。

(3) 透明度:使图像的部分变为透明。

(4) 附加:添加形状或文字到图像。

(5) 装饰:给图像添加边框或帧。

(6) 特效:模糊、锐化和阈值化图像。

(7) 动画效果:可从一组图像中去创建一个 GIF 动画序列。

(8) 文本及评论:插入描述性或艺术性文字到图像中。

(9) 图像识别:能描述出图像的格式和属性。

(10) 组合:重叠一个又一个的图像。

(11) 蒙太奇:并列图像画布上的图像缩略图。

(12) 电影支持:读写图像共同使用的数字电影工作方式。

(13) 图像计算器:能把数学表达式应用到图像或图像通道。

(14) 离散傅里叶变换:实现正向和反向的 DFT。

(15) 加密或解密图片:把图像转换成乱码,然后再转换回普通图像。

(16) 执行:ImageMagick 的是线程安全的,利用内部算法 OpenMP 的功能及快速的双核和四核处理器技术提供窗口优势。

(17) 异构分布式处理:某些算法可以跨越 CPU 和 GPU 以及其他处理器组成的异构平台,这使执行速度大大提高。

(18) 运行在 iPhone:其可以运行在 iOS 系统上对图像进行转换、编辑和组合等操作。

ImageMagick 几乎可以在任何非专有的操作系统上编译,无论是 32 位还是 64 位的 CPU,包括 Linux、Windows 95/98/ME/NT 4.0/2000/XP、Macintosh(MacOS 9/10)、VMS 和 OS/2 等操作系统。

11.1.3 GraphicsMagick

GraphicsMagick 号称图像处理领域的瑞士军刀,其短小精悍的代码却提供了高效的的工具和库来处理图像的读取、写入和操作,支持超过 88 种图像格式,包括重要的 DPX、GIF、JPEG、JPEG-2000、PNG、PDF、PNM 和 TIFF。GraphicsMagick 可以在绝大多数的平台上使用,Linux、Mac 和 Windows 都没有问题。GraphicsMagick 支持大图片的处理,并且已经做过 GB 级别的图像处理实验,同时,还能够动态地生成图片,特别适用于互联网的应用。其可以用来处理调整尺寸、旋转、加亮、颜色调整和增加特效等方面的处理。GaphicsMagick 不仅支持命令行的模式,同时也支持 C、C++、Perl、PHP、Tcl 和 Ruby 等

开发语言的调用。事实上，GraphicsMagick是从ImageMagick 5.5.2分支出来的，但是现在它变得更稳定和优秀。

11.1.4 GD2扩展库、ImageMagick和GraphicsMagick的比较

ImageMagick是目前最流行的图片处理工具，它的功能非常丰富。GraphicsMagick的功能略逊于ImageMagick，但是它的效率更高，就好比Apache和Nginx一样，一个功能更强，一个效率更胜。下面对ImageMagick与GD处理图片的优劣进行比较。

(1) 用Imagemagick替换GD进行图片的处理，相对GD库来说，其功能较强大。

(2) 在画质方面，ImageMagick优于GD。一张像素为1400×1050的带有蓝色天空的图像经ImageMagick、GD1和GD2重新采样转换为像素为600×450的图片，ImageMagick转换的效果最好，GD2次之，GD1最差。网站http://qdig.sourceforge.net/Main/ImageMagickVersusGD对此进行了比较分析。

(3) ImageMagick比GD2具有更优的性能和占用更小的内存。要想了解更详细的情况，可到网站http://www.360doc.com/content/11/0806/22/4171006_138562008.shtml参看结果。

关于性能比较，其实还是存在一点争议。有的测试结果指出，使用某些ImageMagick的API，处理速度反倒不如GD。有的测试结果也指出，少量图片的处理GD的处理速度比ImageMagick要快，随着图片数量和大小的上升，ImageMagick基本不受影响。但是无论如何，在性能方面，ImageMagick有一个重要的优势：因为GD是PHP的一个模块，当PHP接到请求后，其先初始化资源，响应过后需释放资源，所以在大负载的情况下，GD无疑会拖慢PHP程序；而ImageMagick和PHP程序完全是松耦合的关系，如果PHP程序使用命令行调用ImageMagick，那么甚至可以说ImageMagick和PHP程序没什么联系，ImageMagick是通过享用OS的资源给PHP程序提供了服务。如果应用只处理简单的图形图像，那么建议使用GD库处理；如果应用强调图形图像处理的功能和性能，那么建议使用ImageMagick库；如果强调图形图像处理性能，那么建议使用GraphicsMagick，特别是规模较大的图形图像文件。本章重点讲述ImageMagick类库的使用。

11.2 PHP支持ImageMagick环境搭建

11.2.1 准备工作

首先到网站http://www.imagemagick.org/script/binary-releases.php#windows（或者ftp://ftp.imagemagick.org/pub/ImageMagick/binaries/）下载所需版本的ImageMagick函数库，文中下载的是ImageMagick-6.7.9-5-Q16-windows-dll.exe，这里的Q16表示ImageMagick类库支持16位色，该函数库提供了图形图像处理核心功能。其次，为了能让PHP使用ImageMagick函数库开发图像软件，需到http://pecl.php.

net/package/imagick 下载支持 ImageMagick 的 PHP 扩展文件，或到 http://valokuva.org/outside-blog-content/imagick-windows-builds/080709/下载这样的 PHP 扩展文件，文中下载的是 php_imagick_dyn-Q16.dll 这个文件；如果 PHP 版本是 PHP 5.3.x，则需到 http://valokuva.org/outside-blog-content/imagick-windows-builds/php53/下载。

11.2.2 安装

首先安装 ImageMagick-6.7.9-5-Q16-windows-dll.exe 工具，工具的安装较简单，双击程序对应的图标运行程序并连续单击"下一步"按钮即可。然后把 php_imagick_dyn-Q16.dll 重命名为 php_imagick.dll，再复制到 PHP 安装目录下的 ext 文件夹下，在 PHP 安装目录下的配置文件 php-apache2handler.ini(或 php.ini)中添加以下设置：

```
extension=php_imagick.dll
```

重启 Apache Web 服务器后之后，编写 PHP 程序调用函数 phpinfo 并运行程序，如果浏览器里面有没有 imagick 一栏，则说明 PHP 还不支持 ImageMagick，如果有的话则可以使用 PHP 开发 ImageMagick 程序。

11.3 ImageMagick 实例

11.3.1 创建一个缩略图并显示出来

缩略图是网页上或计算机中图片经压缩处理后的小图，因其小巧，浏览器能更加迅速地装入缩略图，常用于快速浏览，相当于图片文件预览及目录的作用。下面是缩略图范例，例中先创建类 imagick 的实例，实例再对图像 flower.jpg 进行处理，指定宽度为 200 像素，由于没有指定高度，因此系统默认高度和宽度一样，最后显示出图像。

【例 11-3-1】 (代码位置：\11\testThumb.php)

```
<?php
  header('Content-type: image/jpeg');
  $image=new Imagick('flower.jpg');
  //If 0 is provided as a width or height parameter
  $image->thumbnailImage(200,0);      //0表示没有指定图像高度,其宽度和长度相同
  echo $image;                        //显示图像
?>
```

程序运行结果如图 11-3-1 所示。



图 11-3-1 缩略图

11.3.2 创建 GIF 缩略图

下面是创建 GIF 缩略图的范例。例中读出文件 flower.jpg，改变其像素为 50×50，并保存为文件 flower_new.jpg。如果把例中的 jpg 文件更换成 gif 文件，那么 gif 动画中的每帧图

片像素都将被修改为 50×50。

【例 11-3-2】（代码位置：\11\testCreateGifThumb.php）

```
<?php
  $im=new Imagick("flower.jpg");
  foreach($im as $frame){
  //每帧图片像素 50×50
  $frame->thumbnailImage(50,50);
  //设置虚拟画布为合适的尺寸
  $frame->setImagePage(50,50,0,0);
  }
  //注意：使用 writeImages 而不是 writeImage
  $im->writeImages("flower_new.jpg",true);
?>
```

11.3.3 验证码

所谓验证码，就是将一串随机产生的数字、字母、符号或文字生成一幅图片，图片里加上一些干扰像素，由用户肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。验证码功能可用于防止批量注册等应用，不少网站为了防止用户利用机器人自动注册、登录和灌水，都采用了验证码技术。常见的验证码有如下几种：

（1）纯数字验证码，一般为 4 位随机数字。

（2）数字＋字母验证码，一般从数字（0～9）和字母（A～Z 和 a～z）中随机抽出几个字符组成。

（3）汉字验证码，一般随机抽出几个汉字组成，相对而言，其实现起来相对复杂一些，但在不少网站中还是可以看到的。

（4）除上述几种验证码，还有带数学计算功能和动画效果的验证码等。

下面是数字＋字母验证码的范例。

【例 11-3-3】（代码位置：\11\testIdCode.php）

```
<?php
  //创建 Imagick 对象
  $Imagick=new Imagick();

  //创建 ImagickPixel 对象(该对象用于设置图像的背景颜色)
  $bg=new ImagickPixel();

  //设置背景为白色
  $bg->setColor('white');

  //创建绘图对象并设置字体尺寸
```

```
    $ImagickDraw=new ImagickDraw();

    //设置字体和字体尺寸,也可通过文件/path/to/font.ttf指定
    $font="c:\\windows\\fonts\\simhei.ttf";
    $ImagickDraw->setFont($font);
    $ImagickDraw->setFontSize(20);

    //创建文本
    $alphanum='ABXZRMHTL23456789';
    $string=substr(str_shuffle($alphanum),2,6);

    //新建空图像
    $Imagick->newImage(85,30,$bg);

    //把文本写到图像中
    $Imagick->annotateImage($ImagickDraw,4,20,0,$string);

    //加入漩涡效果
    $Imagick->swirlImage(20);

    //随机创建线条
    $ImagickDraw->line(rand(0,70),rand(0,30),rand(0,70),rand(0,30));
    $ImagickDraw->line(rand(0,70),rand(0,30),rand(0,70),rand(0,30));
    $ImagickDraw->line(rand(0,70),rand(0,30),rand(0,70),rand(0,30));
    $ImagickDraw->line(rand(0,70),rand(0,30),rand(0,70),rand(0,30));
    $ImagickDraw->line(rand(0,70),rand(0,30),rand(0,70),rand(0,30));

    //把ImagickDraw对象内容画到image中
    $Imagick->drawImage($ImagickDraw);

    //指定图像输出格式
    $Imagick->setImageFormat('png');

    //Send headers and output the image
    header("Content-Type: image/{$Imagick->getImageFormat()}");
    echo $Imagick->getImageBlob();
?>
```

例中生成的验证码效果如图11-3-2所示：

图 11-3-2　验证码

11.3.4　直方图

直方图(histogram)又称柱状图,一般用横轴表示数据类型,纵轴表示分布情况,直方图用来显示统计数据的分布情况。要统计灰度图像的像素灰度分布情况,可以借助直方

图来完成。下面是直方图的范例。

【例 11-3-4】 (代码位置：\11\testHist.php)

```
<?php
  $file='flower.jpg';
  $size=array(
      'width'=>256,
      'height'=>100,
  );
  $image=new Imagick($file);
  $histogram=array_fill_keys(range(0,255),0);
  foreach($image->getImageHistogram()as $pixel){
      $rgb=$pixel->getColor();
      $histogram[$rgb['r']] +=$pixel->getColorCount();
      $histogram[$rgb['g']] +=$pixel->getColorCount();
      $histogram[$rgb['b']] +=$pixel->getColorCount();
  }
  $max=max($histogram);
  $threshold=($image->getImageWidth() * $image->getImageHeight())/256 * 12;
  if($max >$threshold){
     $max=$threshold;
  }
  $image=new Imagick();
  $draw =new ImagickDraw();
  $image->newImage($size['width'],$size['height'],'white');

  foreach($histogram as $x=>$count){
      if($count==0){
          continue;
      }
      $draw->setStrokeColor('black');
      $height=min($count,$max)/ $max * $size['height'];
      $draw->line($x,$size['height'],$x,$size['height'] -$height);
      $image->drawImage($draw);
      $draw->clear();
  }
  $image->setImageFormat('png');
  $image->writeImage('histogram.png');
  echo $image;
?>
```

例中对原图(flower.jpg)生成直方图，运行结果如图 11-3-3 所示，其中，图 11-3-3(a)是原图，图 11-3-3(b)是原图的像素灰度分布直方图。


(a) 原图


(b) 像素灰度分布直方图

图 11-3-3　原图和像素灰度分布直方图

11.3.5　图像合并

有的时候要把多张图片拼成一张图做成各种效果。下面是图像合并的范例。

【例 11-3-5】（代码位置：\11\testMergePics. php）

```
<?php
  $canvas=new imagick("flower.jpg");
  $canvas->adaptiveResizeImage(300,300);
  $canvas->setImageFormat("png");
  $paths=array("flower1.jpg","flower2.jpg");
  //创建一个空的 ImagickDraw 对象(使用默认设置,偏振)
  $bg=new ImagickDraw();
  //创建一些随机图片
  $images=new Imagick($paths);
  //循环图像,覆盖在该画布上
  foreach($images as $key=>$image)
  {
    //缩略图宽 100,设置背景颜色为白色
    $image->thumbnailImage(100,null);
    $image->setImageBackgroundColor(new ImagickPixel("black"));
    //使用一个随机角度
    $angle=mt_rand(1,45);
    if(mt_rand(1,2)%2===0)
    {
        $angle=$angle * -1;
    }
    //创建偏振
    $image->polaroidImage($bg,$angle);
    //在画布上组合,并移动到随机位置
    $canvas->compositeImage($image,Imagick::COMPOSITE_OVER,mt_rand(10,150),mt_rand(10,
    150));
    //释放
    $image->removeImage();
  }
  header("Content-Type: image/png");
```

```
    echo $canvas;
?>
```

例中把3张图片合并起来，其中两张图片以随机产生的角度、摆放位置和指定尺寸合并到另一张图片，运行结果如图11-3-4所示。



图 11-3-4　合并效果

11.3.6　字母和数字生成png图片

有时候需要在贺卡画片上签上留言和时间等文字内容，下面是把字母和数字生成png图片的范例。



【例 11-3-6】（代码位置：\11\testCardImg.php）

```
<?php
  function text($text,$color,$size,$font,$type=false,$src='')
  {
      $font="c:\\windows\\fonts\\" . $font . ".ttf";
      $draw=new ImagickDraw();
      $draw->setGravity(Imagick::GRAVITY_CENTER);
      $draw->setFont($font);
      $draw->setFontSize($size);
      $draw->setFillColor(new ImagickPixel($color));
      $im=new imagick();
      $properties=$im->queryFontMetrics($draw,$text);
      $im->newImage(intval($properties['textWidth'] + 5),intval($properties['
      textHeight']+5),new ImagickPixel('transparent'));
      $im->setImageFormat('png');
      $im->annotateImage($draw,0,0,0,$text);
      if($type)
      {
          $im->writeImage($src);
          return $src;
      }
      else return $im;
  }

  $image=text("0123456789演示","RGB(200,0,0)","100","simhei");
  echo $image;
?>
```

例中函数text的功能是将字母和数字生成png图片，参数$text表示需要生成图片的文字，其类型为string型；参数$color为文字颜色，其为string类型，其值可以是red等颜色名或RGB(0-255,0-255,0-255)，参数$szie为文字大小，其为int型；参数$font为字体，其类型为string型；参数$type为返回类型，其为逻辑型，当值为true时表示返

回图片地址,值 false 表示返回图片资源;参数 $ src 为保存图片的地址,为 string 型。程序运行后会生成图片并在图中以红色和黑体字显示“0123456789 演示”。

11.3.7 多边形

由 3 条或 3 条以上的线段首尾顺次连接所组成的封闭图形叫做多边形,例如星形图形。要绘制不规则的图形,可以借助数组描述多边形的特征点,并借助函数 polygon 自动绘制多边形。下面是多边形绘制的范例。

【例 11-3-7】 (代码位置:\11\testPolygon. php)

```
<?php
  //创建 Imagick 对象
  $Imagick=new Imagick();
  //创建 ImagickDraw 对象
  $ImagickDraw=new ImagickDraw();
  //创建 ImagickPixel 对象
  $ImagickPixel=new ImagickPixel();
  //数组包含多边形几何坐标
  $array=array (array("x"=>378.1,"y"=>81.72),
              array("x"=>381.1,"y"=>79.56),
              array("x"=>384.3,"y"=>78.12),
              array("x"=>387.6,"y"=>77.33),
              array("x"=>391.1,"y"=>77.11),
              array("x"=>394.6,"y"=>77.62),
              array("x"=>397.8,"y"=>78.77),
              array("x"=>400.9,"y"=>80.57),
              array("x"=>403.6,"y"=>83.02),
              array("x"=>523.9,"y"=>216.8),
              array("x"=>526.2,"y"=>219.7),
              array("x"=>527.6,"y"=>223),
              array("x"=>528.4,"y"=>226.4),
              array("x"=>528.6,"y"=>229.8),
              array("x"=>528.0,"y"=>233.3),
              array("x"=>526.9,"y"=>236.5),
              array("x"=>525.1,"y"=>239.5),
              array("x"=>522.6,"y"=>242.2),
              array("x"=>495.9,"y"=>266.3),
              array("x"=>493,"y"=>268.5),
              array("x"=>489.7,"y"=>269.9),
              array("x"=>486.4,"y"=>270.8),
              array("x"=>482.9,"y"=>270.9),
              array("x"=>479.5,"y"=>270.4),
              array("x"=>476.2,"y"=>269.3),
              array("x"=>473.2,"y"=>267.5),
```

```
            array("x"=>470.4,"y"=>265),
            array("x"=>350,"y"=>131.2),
            array("x"=>347.8,"y"=>128.3),
            array("x"=>346.4,"y"=>125.1),
            array("x"=>345.6,"y"=>121.7),
            array("x"=>345.4,"y"=>118.2),
            array("x"=>346,"y"=>114.8),
            array("x"=>347.1,"y"=>111.5),
            array("x"=>348.9,"y"=>108.5),
            array("x"=>351.4,"y"=>105.8),
            array("x"=>378.1,"y"=>81.72),
           );

  //设置背景颜色
  $ImagickPixel->setColor('red');
  //创建新图像,图像颜色为灰色,图像格式为 png
  $Imagick->newImage(700,500,$ImagickPixel);
  $Imagick->setImageFormat('png');
  //创建多边形
  $ImagickDraw->polygon($array);
  //渲染多边形到图片中
  $Imagick->drawImage($ImagickDraw);
  //设置页面头信息,声明页面为图像
  header("Content-Type: image/{$Imagick->getImageFormat()}");
  echo $Imagick->getImageBlob();
?>
```

范例运行结果如图 11-3-5 所示,其中,黑色倾斜的带有拐弯的方框为多边形。

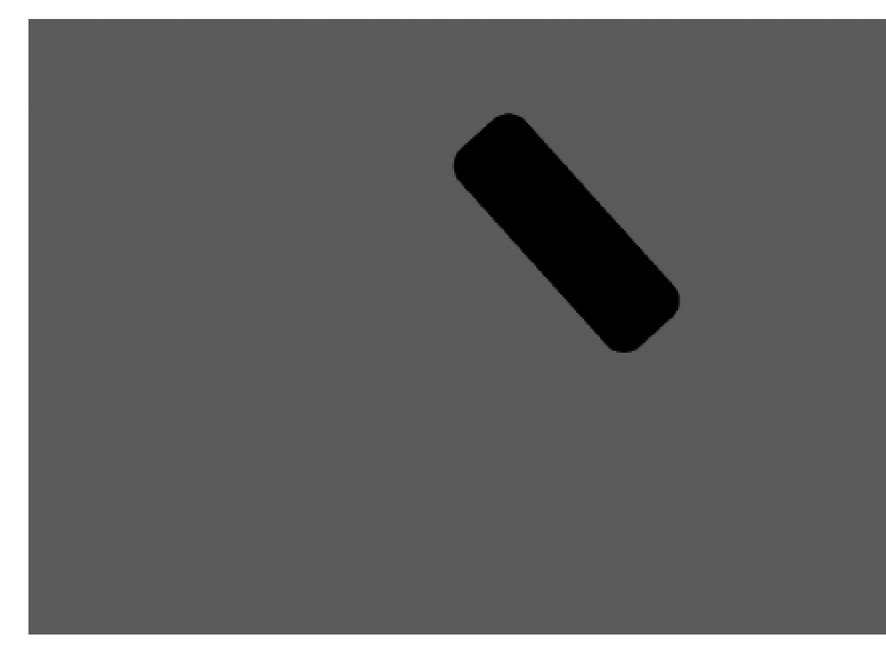


图 11-3-5 多边形

11.3.8 像素操作

通常,图像处理工作都是诸如缩略图、水印之类的简单工作,不过有的时候为了实现自定义的图片效果,需要对图片进行像素级读写操作,可以通过像素迭代来完成像素操作。下面是写像素操作的范例。

【例 11-3-8】 (代码位置:\11\testPixel.php)

```
<?php
  $data=array();
  for($row=0;$row<100;$row++)
  {
    for($column=0;$column<100;$column++)
    {
```

```
        $data[$row][$column]='#' . str_repeat($column %10,6);
      }
    }
    $image=new Imagick();
    $image->newimage(100,100,'white','png');
    $iterator=$image->getPixelIterator();        //获取像素迭代器
    foreach($iterator as $row=>$pixels)
    {
      foreach($pixels as $column=>$pixel)
      {
        $pixel->setColor($data[$row][$column]);
      }
    $iterator->syncIterator();                   //写同步操作
    }
    echo $image;
?>
```

注意：在Imagick中利用PixelIterator写像素时，需要调用syncIterator操作(读像素不用)。从例中可知，当进行写像素操作时使用了函数setColor，当读像素时可使用函数getColor。范例运行结果如图11-3-6所示。

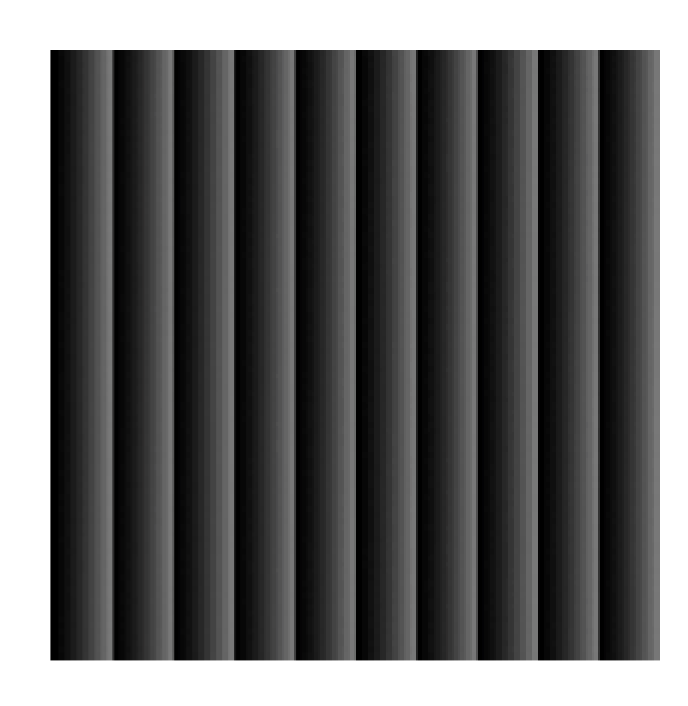


图11-3-6　写像素

第12章 chapter 12

PHP经典项目

12.1 投 票 系 统

12.1.1 投票系统简述

当出现了一个热门话题，例如“您对2013年春节晚会有什么看法”时，使用新闻模块能够进行网站和用户之间的交流，但是却很难明显地看出用户的意愿，也无法统计用户的观点信息。使用投票功能能够良好地解决这个问题，投票能够很好地进行统计，非常直观地呈现百分比。在一些网站的应用中，为了加强用户和网站之间的交互，常常开发投票模块让用户能够参与到网站的活动中，网站还能够通过投票模块进行用户信息的统计和调查，使用投票模块能够更好地让网站与用户进行交互。投票模块能够加强用户和网站信息之间的互动，网站可以使用投票功能进行网站内容的调查，例如调查用户是否满意网站的设计或者是否满意网站改版，等等。同时投票模块还能够进行热点事件的调查，例如“你怎么看待××事件”等，都可以使用投票模块进行统计。

12.1.2 关键技术

投票系统的关键实现包括动态生成发布投票界面、动态生成投票界面以及统计显示选票结果。显示选票统计结果可以用JpGraph类库绘制成柱形图，下面介绍JpGraph类库。

JpGraph这个强大的绘图组件能根据用户的需要绘制任意图形。只需要提供数据，就能自动调用绘图函数的过程，把处理的数据输入自动绘制。JpGraph提供了多种方法创建各种统计图，包括折线图、柱形图和饼形图等。JpGraph是一个完全使用PHP语言编写的类库，并可以应用在任何PHP环境中。JpGraph可以从其官方网站http://jpgraph.net下载。JpGraph支持PHP 4.3.1以上和PHP 5两种版本的图形库，可以选择合适的JpGraph下载。目前JpGraph最新的版本是3.5.0b1，对应的文件名是jpgraph-3.5.0b1.tar.gz，JpGraph 3.x只支持PHP 5或更高的版本，不适用于PHP 4或更低的版本。

JpGraph需要GD库的支持，使用JpGraph前需要开启GD扩展库。JpGraph的安

装其实就是将从官方网站下载的文件 jpgraph-3.5.0b1.tar.gz 解压复制到 PHP 程序可以找到的目录中。PHP 应用程序要使用 JpGraph 类库，需要应用语句 include 引用文件 jpgraph.php 和 jpgraph_bar.php，其中 jpgraph_bar.php 是柱形图形类文件，用来创建矩形图形。创建柱形图形的过程是：首先要创建画布，设置 X 和 Y 坐标轴文字以及其他属性值；然后创建矩形，设置矩形数字等属性；最后把矩形添加到画布中并显示出来。JpGraph 图片显示在网页中的方法：生成图片的 php 文件只能产生和返回图像，在 JpGraph 生成图像的代码之前不要有任何 HTML 输出；在 HTML 中显示 JpGraph PHP 文件产生图像的唯一方法是使用<img>标志，使用时设置 src 属性为 PHP 文件；在同一 HTML 文件中显示多个 JpGraph 图像的方法就是使用多个<img>标签。

如果用户希望 JpGraph 类库仅对当前站点有效，只需将 JpGraph 压缩包下的 src 文件夹中的全部文件复制到网站所在目录的文件夹中即可，使用时调用 src 文件夹下的指定文件即可。

12.1.3 系统需求

投票系统可以快速地统计群体对某观点的意向，投票系统有管理员和用户两类使用者，以及拥有发布投票、删除选票、投票和查看票数 4 种功能。

(1) 发布投票功能用于管理员发布投票，管理员可以在投票发布页面填写投票信息，一项投票信息包括一项投票主题和多项投票选项，管理员需要在界面文本框中填写多项投票选项，每一行表示一个选项。

(2) 删除选票功能用于管理员删除无效的选票，包括删除选票主题和相关的选项。

(3) 管理员发布投票后，用户就能够在前台进行相应的操作，包括投票和查看投票票数等，投票模块只允许用户就一项投票主题选择一个投票选项，查看票数功能以文本和图形两种形式向用户显示投票票数和各选项百分率。

(4) 除了功能性需求外，系统还需要非功能性需求用来提高系统的可靠性等，包括防刷新功能等。

为了简单起见，系统没有加入用户登录和修改选票等功能。投票系统用例图如图 12-1-1 所示。



图 12-1-1 投票系统用例图

12.1.4 数据库设计

一张投票由一个投票主题和多个投票选项一块进行描述。可以用一个数据库投票主题表管理投票主题，投票主题表用于存放和投票相关的问题的数据，投票主题表要包括投票编号、投票票题和投票内容等字段。用另一个数据库投票选项表管理投票选项，

投票选项表用于存放和投票选项相关的数据，投票选项表要包括投票选项编号、投票选项统计、投票描述和投票编号等字段。投票编号用来关联投票主题和投票选项，从而能够共同描述投票。投票主题表和投票选项表具体如表 12-1-1 和表 12-1-2 所示。

表 12-1-1　投票主题表 votesubject

字　　段	字段名称	数据类型	字 段 含 义	备　　注
vote_subject_id	投票编号	int(10)	用于标识投票主题，为自动增长的主键	AUTO_INCREMENT
vote_subject_title	投票标题	varchar(80)	用于显示标题，作为投票模块的标题	
vote_subject_text	投票内容	text	用于解释投票信息的一些内容	

表 12-1-2　投票选项表 vote

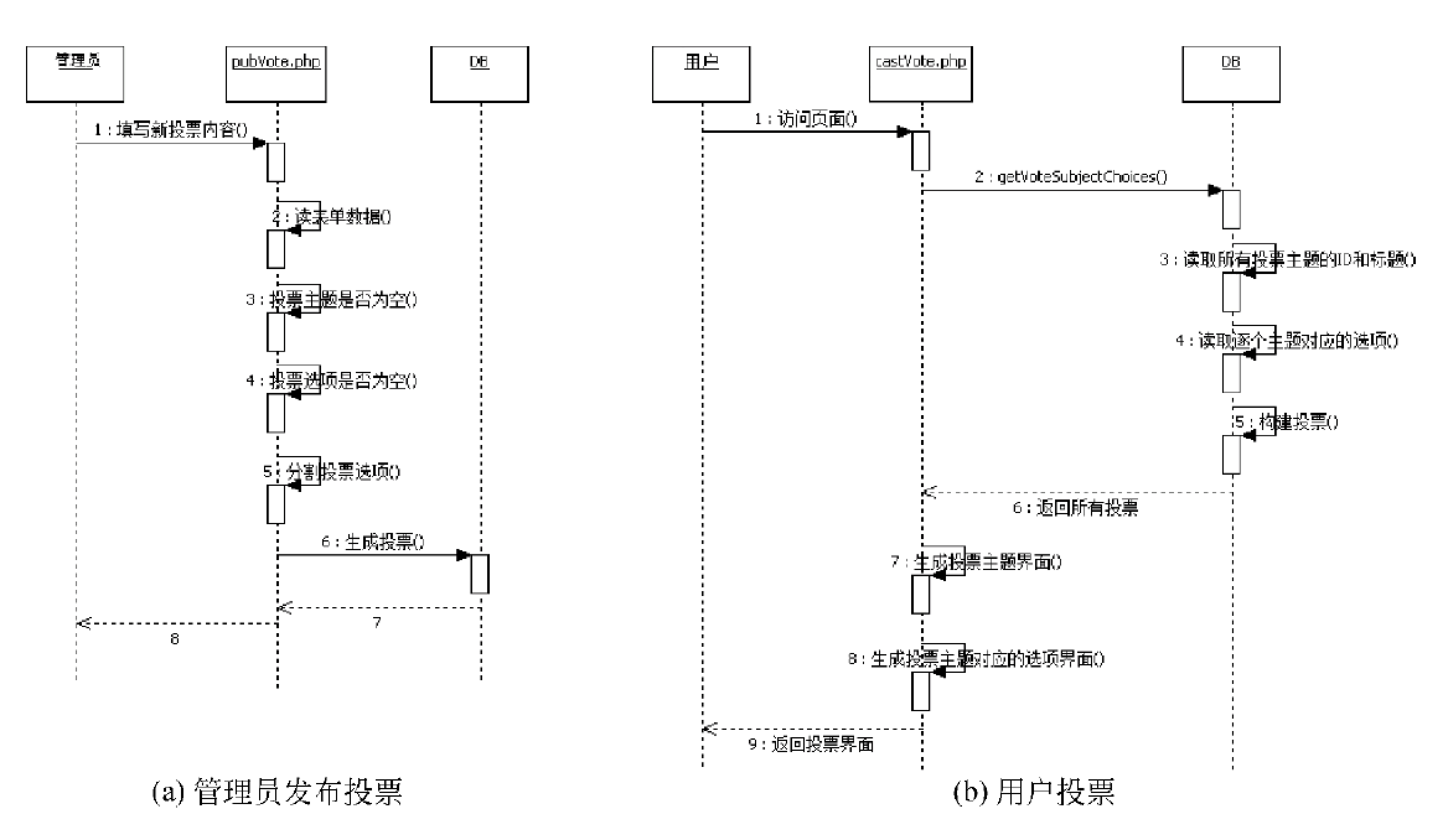
字　　段	字段名称	数据类型	字 段 含 义	备　　注
voteid	投票选项编号	int(10)	用于标识投票选项，为自动增长的主键	AUTO_INCREMENT
vote_counts	投票选项统计	int(50)	用于标识该投票被选择的次数	
vote_text	投票描述	text	用于描述投票中的一个选项	
vote_subject_id	投票编号	int(10)	用于标识该投票选项是隶属于哪个投票问题，为投票问题表的外键	

12.1.5　系统设计和文件结构

管理员发布投票处理过程：管理员先填写投票内容并提交给文件 pubVote. php 处理，投票内容包括投票主题和选项；然后读出用户填写的投票内容，判断投票主题和选项是否为空，如果为空则给出错误提示和结束业务处理，否则把投票选项按行分割出若干选项，并把投票主题和分割后的选项提交给类 DB 生成新投票；类 DB 把投票主题和选项分别存到数据库中的投票主题表和投票选项表中。

用户投票处理过程如下：用户访问页面 castVote. php；castVote. php 请求类 DB 读出所有投票，类 DB 先从投票主题表中读出所有投票主题编号和标题，然后按投票主题逐个读出投票选项，最后把投票主题和对应选项组成投票；castVote. php 逐个处理投票生成选票界面，界面显示有投票主题、投票选项内容和投票选择。管理员发布投票过程和用户投票过程如图 12-1-2 所示，其中，图 12-1-2(a)是管理员发布投票过程，图 12-1-2(b)是用户投票过程。

系统由 db. php、pubVote. php、castVote. php、resultVote. php、graphResult. php 和 vote. css 等构成。文件 db. php 用来操作数据库，数据库采用面向对象编程，设计有投票主题类、投票选项类和 DB 类，主要操作包括从数据库中读取投票、插入投票和投票计数等。文件 pubVote. php 是管理员发布投票界面，具有数据检验和发布投票等功能。文件



(a) 管理员发布投票　　(b) 用户投票

图 12-1-2　投票系统设计

castVote. php 是用户投票界面，它能动态生成投票界面。文件 resultVote. php 是查看票数界面，用来收集用户的投票、统计投票选项和显示投票数，文件 graphResult. php 以图形方式显示投票结果。vote. css 是 CSS 文件，用来设计管理员使用的发布投票界面和用户使用的投票界面。

12.1.6　系统实现

下面是数据库类的范例，例中拥有 Votesubject、Vote 和 DB 类，Votesubject 表示投票主题类，Vote 表示选项类，DB 类用来操作数据库，包括把选票插入到数据库等操作。

【例 12-1-1】（代码位置：\12\vote\db. php）

```
<?php
class Votesubject
{ //投票主题类
    public $vote_subject_id;              //投票主题编号
    public $subjecttitle;                 //投票主题的标题
    public $votes;                        //投票主题的若干选项
}
class Vote
{ //选项类
    public $voteid;                       //选项编号
    public $vote_text;                    //选项内容说明
    public $vote_counts;                  //选项投票数
}
class DB
{ //数据库操作类
```

```
public function countVote($vote_subject_id,$voteid)
{
    //功能：票计数
    //$vote_subject_id: 投票主题编号
    //$voteid: 选项编号
    $con=mysql_connect("localhost","root","111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }

    mysql_select_db("votesys",$con);
    mysql_query("set names utf8");      //让数据库支持中文
    mysql_query("UPDATE vote SET vote_counts=vote_counts+1
    WHERE voteid='$voteid' AND vote_subject_id='$vote_subject_id'");
    mysql_close($con);
}
public function getVoteSubjectIds()
{
    //功能：读取所有投票主题编号
    //返回值：以数组形式返回投票主题编号
    $result=array();
    $con=mysql_connect("localhost","root","111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("votesys",$con);
    mysql_query("set names utf8");      //让数据库支持中文
    $resultvotesubject=mysql_query("SELECT distinct * FROM votesubject");
    while($row=mysql_fetch_array($resultvotesubject))
    {
        $result[]=$row['vote_subject_id'] ;
    }
    mysql_close($con);
    return $result;
}
public function getVoteSubjectChoices()
{
    //功能：读取所有投票主题和对应选项
    //返回值：以对象数组形式返回投票主题和选项
    $result=array();
    $con=mysql_connect("localhost","root","111111");
    if(!$con)
```

```
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("votesys",$con);
        mysql_query("set names utf8");     //让数据库支持中文
        $resultvotesubject=mysql_query("SELECT distinct * FROM votesubject");
        //扫描每一个投票主题
        while($rowvotesubject=mysql_fetch_array($resultvotesubject))
        {
            $votesubject=new Votesubject();
            $votesubject->vote_subject_id=$rowvotesubject['vote_subject_id'];
            $votesubject->subjecttitle=$rowvotesubject['vote_subject_title'];
            $votesubject->votes=array();
            $vote_subject_id=$rowvotesubject['vote_subject_id'] ;
            $resultvote=mysql_query("SELECT * FROM vote where vote_subject_id=
            '$vote_subject_id' ");
            //扫描投票主题所对应的所有选项
            while($rowvote=mysql_fetch_array($resultvote))
            {
                $vote=new Vote();
                $vote->voteid=$rowvote['voteid'] ;
                $vote->vote_text=$rowvote['vote_text'] ;
                $vote->vote_counts=$rowvote['vote_counts'] ;
                $votesubject->votes[]=$vote;
            }
            //投票主题和选项保存到数组中
            $result[]=$votesubject;
        }
        mysql_close($con);
        return $result;
    }
    public function newVoteSubjectChoices($subjecttitle,$subjecttext,$voteChoices)
    { //功能:插入投票
      //$subjecttitle:投票主题
      //$subjecttext:投票说明
      //$voteChoices:投票若干选项,以数组表示
        $con=mysql_connect("localhost","root","111111");
        if(!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("votesys",$con);
        mysql_query("set names utf8");    //让数据库支持中文
        //插入一个选票主题
```

```
            mysql_query("INSERT INTO votesubject(vote_subject_title,vote_subject_text)
            VALUES('$subjecttitle','$subjecttext')");
            //获取该投票主题的 id
            //默认 id=1000000000,表示无效
            $vote_subject_id="1000000000";
            $result=mysql_query("SELECT vote_subject_id FROM votesubject where
                    vote_subject_title='$subjecttitle' and vote_subject_text='
                    $subjecttext' limit 1 ");
            while($row=mysql_fetch_array($result))
                $vote_subject_id=$row['vote_subject_id'];
            //插入若干选票选项
            foreach($voteChoices as $key=>$value)
            {
                mysql_query("INSERT INTO vote(vote_subject_id, vote_text)
                VALUES('$vote_subject_id','$value')");
            }
            mysql_close($con);
        }
}
?>
```

下面是界面样式的范例,用于设置选票主题和选项等项目的样式。

【例 12-1-2】 (代码位置:\12\vote\vote.css)

```
#subjecttitle {
        background-color:#ffe0a0;
}
#voteoption {
        background-color:#fff0f0;
}
#voteselect {
        background-color:#ff8050;
}
```

下面是管理员发布投票的范例。管理员可通过访问文件 pubVote.php 来发布投票。

【例 12-1-3】 (代码位置:\12\vote\pubVote.php)

```
<html>
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <title>发布投票</title>
</head>
<body>
<form method="post" action="pubVote.php">
  投票主题:<input type="text" name="subjecttitle" />
  <br/><br/>
```

```
  投票选项(<font color="red" size="2px">注：每一行代表一个选项</font>)：<br/>
  <textarea rows="10" cols="30" name="subjecttext"></textarea>
  <input type="submit" value="提交"/>
</form>
<?php
include_once("db.php");
$subjecttitle=$_REQUEST["subjecttitle"];
$subjecttext=$_REQUEST["subjecttext"];
if((isset($subjecttitle)==false)||(isset($subjecttext)==false))
{
    echo "<script type='text/javascript'>alert('获取投票主题和选项出错')</script>";
    return;
}
if((trim($subjecttitle)=="")||(trim($subjecttext)==""))
{
    echo "<script type='text/javascript'>alert('投票主题和选项不能为空')</script>";
    return;
}
$voteChoices=explode("\n",$subjecttext);
$db=new DB();
$db->newVoteSubjectChoices($subjecttitle,$subjecttext,$voteChoices);

?>
</body>
</html>
```

运行该程序后，管理员可先在投票主题和投票选项中填写内容，其中，投票选项用一行表示一个选项；然后单击“提交”按钮发布投票；最后程序会在数据库中插入投票记录。其运行结果如图 12-1-3 所示。

投票主题：公司最优秀员工？

投票选项（注：每一行代表一个选项）：

张小三
李四
王老五
陈小双

提交

图 12-1-3 发布投票

由于管理员会随时添加新的投票，所以用户投票的界面要动态生成。下面是用户投票的范例，例中文件 castVote.php 用来动态添加投票主题和选项输出投票界面。

【例 12-1-4】 （代码位置：\12\vote\castVote.php）

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="vote.css"/>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <title>投票</title>
</head>
<body>
<form method="post" action="resultVote.php" >
```

```
<?php
//文件功能:生成投票界面供用户投票
include_once("db.php");
$db=new DB();
$votesubjects=$db->getVoteSubjectChoices();
foreach($votesubjects as $key=>$subject)
{
    $vote_subject_id=$subject->vote_subject_id;
    $subjecttitle=$subject->subjecttitle;
    $votes=$subject->votes;
    //输出投票主题
    echo "<div id='subjecttitle'>$subjecttitle</div>";
    echo "<div id='voteoption'>";
    //输出该投票主题的选项
    foreach($votes as $votekey=>$vote)
    {
        $vote_text=$vote->vote_text;
        $vote_counts=$vote->vote_counts;
        echo "选项$votekey: $vote_text $vote_counts 票 <br/>";
    }
    echo "</div>";
    echo "<div id='voteselect'>";
    echo "您的选项:";
    echo "<select name='$vote_subject_id"."[]' >";
    foreach($votes as $votekey=>$vote)
    {
        $voteid=$vote->voteid;
        echo "<option value='$voteid' >$votekey</option>";
    }
    echo "</select>";
    echo "</div>";
    echo "----------------------------------------------<br/>";
}
?>
<input type="submit" value="提交"/>
</form>
</body>
</html>
```

程序运行后会从数据库中读出所有投票并生成投票界面,界面中的每一项投票都包含投票主题、投票选项和用户选项下拉列表3项内容,用户可在用户选项下拉列表中选择自己想投的选项。程序运行结果如图12-1-4所示。

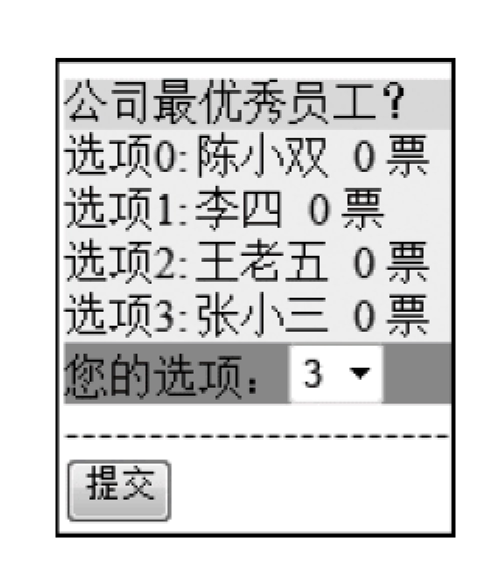


图12-1-4　投票界面

动态输出投票界面后,用户就可以进行投票了。用户投票后,系统要能记录下投票、统计投票选项结果和以图形方式显示投票结

果。下面是用户查看票数的范例，查看票数模块由 resultVote. php 和 graphResult. php 两个文件构成。文件 resultVote. php 用来记录投票和统计投票结果。文件 graphResult. php 的作用是依据投票结果输出柱形图形。文件 resultVote. php 以 URI 方式把投票统计结果传递给文件 graphResult. php 处理生成直方图统计图形，传递的格式是 optsx1＝×&…&optsxn＝×，例如，opts1＝90. 00&opts2＝10. 00 表示有两个选项，选项 1 得票率为 90％，选项 2 得票率为 10％。范例实现代码如下。

【例 12-1-5】（代码位置：\12\vote 目录）

文件 resultVote. php 内容如下：

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="vote.css"/>
    <meta http-equiv="content-type" content="text/html;charset=utf-8"/>
    <title>统计票数</title>
</head>
<body>
<?php
include_once("db.php");
include_once("jpgraph/jpgraph.php");
require_once('jpgraph/jpgraph_bar.php');
$db=new DB();
//投票选项计数加一处理
$subjectids=$db->getVoteSubjectIds();
foreach($subjectids as $key=>$subjectid)
{
    $id=$_REQUEST[$subjectid];
    $db->countVote($subjectid,$id[0]);          //选项票数加一
}
//统计所有投票选项的得票数
$votesubjects=$db->getVoteSubjectChoices();
foreach($votesubjects as $key=>$subject)
{
    $vote_subject_id=$subject->vote_subject_id;
    $subjecttitle=$subject->subjecttitle;
    $votes=$subject->votes;
    echo " <div id='subjecttitle'>$subjecttitle </div> ";
    echo "<div id='voteoption'>";
    $opts=array();                              //每个选项得票率
    $count=0;                                   //选票总数
    $item=1;
    foreach($votes as $votekey=>$vote)
    {
        $vote_text=$vote->vote_text;
```

```
        $vote_counts=$vote->vote_counts;
        echo "选项$item: '$vote_text' 得票数为$vote_counts<br/>";
        $opts[]=$vote_counts;                   //记录每项投票数
        $count+=$vote_counts;                   //计算总票数
        $item++;
    }
    //计算选项得票率
    $optsURI="";                    //选项得票率以 URI 方式传值给文件 graphResult.php
    $item=0;
    foreach($opts as $key=>$value)
    {
        $rate=$value/$count;                    //选项得票率
        $rate=number_format($rate,2);
        $optsURI.="opts$item=$rate&&";
        $item++;
    }
    //以柱形图形显示投票结果
    echo "<img src='graphResult.php?$optsURI'/>";
    echo "</div>";
    echo "<br/>";
}
?>
</body>
</html>
```

文件 graphResult.php 内容如下：

```
<?php
include("jpgraph/jpgraph.php");
include("jpgraph/jpgraph_bar.php");
$resultes=array();
$vote_content=array();
foreach($_REQUEST as $key=>$value)
{
    if(substr_count($key,"opt")>0)
    {
        $resultes[]=$value;
    }
}
for($i=1;$i<20;$i++)
    $vote_content[]="选项$i";
$graph=new Graph(400,200,"auto");               //创建画布
$graph->SetScale("textlin");
$graph->yaxis->scale->SetGrace(20);
//创建画布阴影
```

```
$graph->SetShadow();
$graph->img->SetMargin(40,30,30,40);          //设置显示区左、右、上、下距边线的距离,单位为像素
$bplot=new BarPlot($resultes);                 //创建一个矩形的对象
$bplot->SetFillColor('red');                   //设置柱形图的颜色
$bplot->value->Show();                         //设置显示数字
$bplot->value->SetFormat('%d');                //在柱形图中显示格式化的评选结果
$graph->Add($bplot);                           //将柱形图添加到图像中
$graph->SetMarginColor("lightblue");           //设置画布背景色为淡蓝色
$graph->title->Set("选项得票率");              //创建标题
$graph->xaxis->SetTickLabels($vote_content);   //设置 X 坐标轴文字
$graph->title->SetFont(FF_SIMSUN);             //设置字体
$graph->xaxis->SetFont(FF_SIMSUN);             //设置 X 坐标轴字体
$graph->Stroke();                              //输出矩形图表
?>
```

程序运行结果如图 12-1-5 所示。

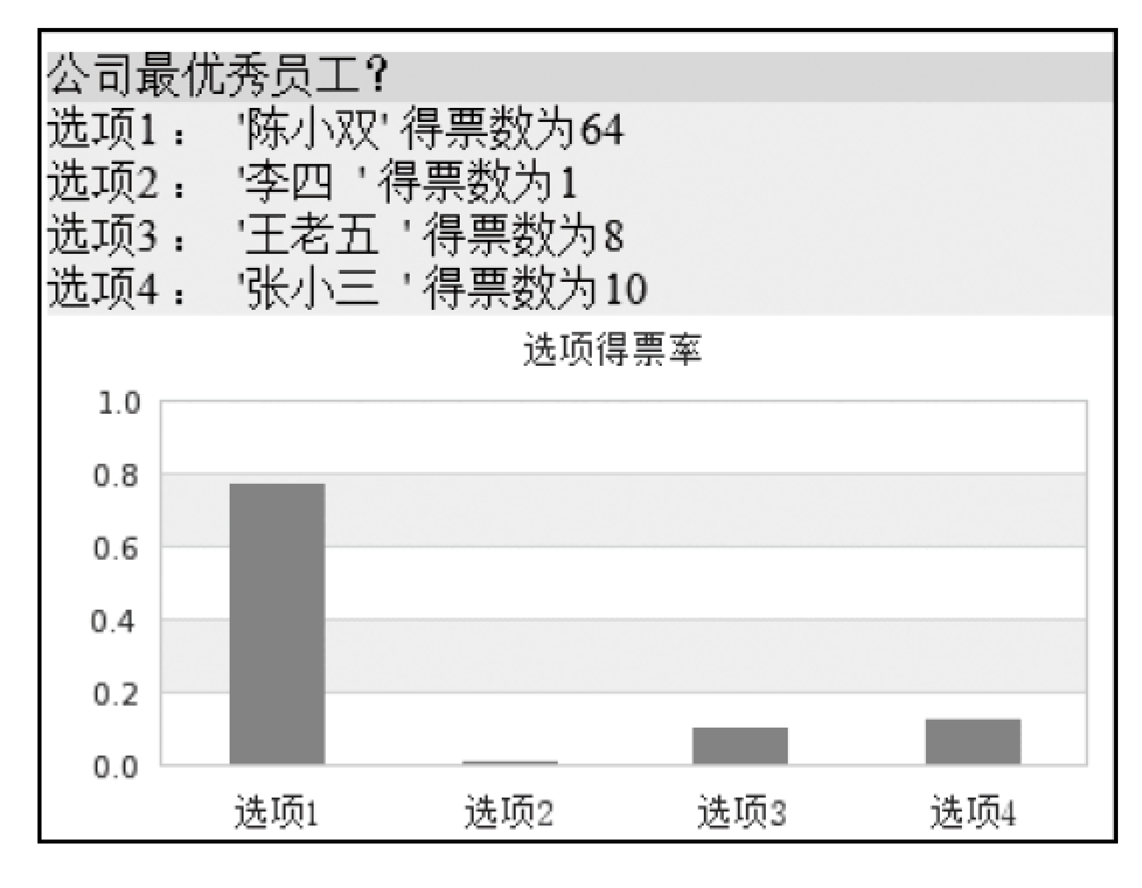


图 12-1-5 投票选项统计结果

12.2 聊天室系统

12.2.1 Web 聊天室及实现原理

随着 Internet 技术的飞速发展,网络已经成为人们生活中不可缺少的一部分,通过聊天室在线聊天已成为网络上人与人之间沟通、交流和联系的一种方式。为此,越来越多的网站开始提供在线聊天的功能。与此同时,聊天室也以其方便、快捷和低成本等优势受到众多企业的青睐,很多企业的网站中也加入了聊天室模块,以达到增进企业与消费者之间、消费者与消费者之间相互交流的目的。聊天室可以建立在 TCP/IP、P2P 和 HTTP 等基础上,基于 HTTP 协议的 Web 聊天方式更为普遍,种类繁多,交谈的手段不局限于文本,还包括图片、语音和视频。通常聊天室是按照房间或频道为单位的,在同一房间或频道的人可以实时地广播和阅读公开消息。一般情况下,与其他网络论坛和即时

通信不同的是，聊天室不保存聊天记录。Web 聊天方式采用浏览器进行聊天，其基本原理是把每个用户的发言通过浏览器传给服务器，再由服务器收集处理后分发给特定用户。

现在有许多 Web 聊天室，它们使用户可以通过浏览器进行聊天，而不必下载专用的客户端。Web 聊天室的优点也在于此，它适用范围广，几乎所有接入 Internet 的用户都装有浏览器。下载专用客户端聊天软件可能是用户比较反感的，因为相对于使用浏览器来说，它们比较烦琐。更重要的是，使用浏览器，能使安全性得到更好的控制，在浏览器中的程序是严格受限制的，它们不会（或很难）做出对用户不利的举动。

Web 聊天一般采用服务器推送（Server Push）或客户端拉（Client Pull）技术，两种技术的区别在于使用不同的方式将数据分发给用户。服务器推送是当客户端新数据到达时由服务器将数据以多重 MIME 编码推（push）给其他客户端。基于服务器推送聊天室的优点是很明显的，它既节约了网络带宽，降低了网络延迟，又减少了服务器的负担。但是，由于 HTTP 协议的限制，服务器端不可能直接发起连接到客户端并传输数据，整个实现空间被完全限制在客户从服务器拉数据的方式中，实现难度大。客户端拉则是客户端主动从服务器拉（pull）所要的数据。最常用的客户端拉就是利用 HTML 语言的 Meta 标签的属性 http-equiv="Refresh"或 JavaScript 脚本的 setTimeout 方法，每隔一段时间就检查服务器上是否有新的数据。例如，每隔 5 秒钟就会刷新一次页面。这类聊天室的优点是实现简单，而且，如果网络延迟不是问题的话，其总体成本将是最优的。但实际上，网络速度经常是一个很重要的问题，而由于这种聊天室网络传输量大，在一个低速的网络上，延迟将变得很明显。再有，频繁的数据传输任务还将加重服务器的负担。本节采用客户端拉技术实现聊天室。

12.2.2 关键技术

聊天室的关键技术包括动态刷新在线用户列表、在线用户列表管理、用户发言和动态刷新聊天内容。在线用户列表和聊天内容数据被所有用户共享，由于用户经常上下线和频繁发言，客户端需要不断地刷新数据以及服务器频繁地读写数据，因此数据读写性能是系统需要解决的一个关键问题。用文件管理数据容易实现，但是多人使用，频繁地对同一文件进行 I/O 操作，难免会影响效率。可以考虑使用 RamDisk，将文件整个放进内存以提高速度，或者在内存中划分出一块高速空间用以数据存放。本节使用的是 MySQL 数据库方式。因为 MySQL 数据库是为大批量用户同时使用而设计的，利用它可以省去设计高速空间操作的复杂代码编写，性能也可得到保证。如果将整个数据库放进内存，那么其性能会更好。

动态刷新在线用户列表和聊天内容比较传统的解决方法是使用框架和刷新的方法，即整个页面被分为显示在线用户列表、发言聊天内容和查看聊天内容 3 个显示区域，每个区域由 HTML 框架指向目标页面。用户在发言聊天内容页面中输入发言内容后发送给服务器，数据经过处理后保存在 MySQL 数据库，同时被保存的还包括发言人、聊天对象及发言时间。在查看聊天内容和在线用户列表页面中，页面按一定间隔时间不断地实现自刷新，每次刷新都从服务器中读出用户列表和发言内容。这样的弊端显而易见，由

于要刷新整个页面，所以会造成一定的屏幕闪烁。而这种情况在聊天数据非常大的情况下表现尤其突出。本节采用 Xajax 可以真正实现无刷新。使用 Xajax 可以只对屏幕中的一部分内容进行更新，而不需要完全载入整个页面，这样可以在很大程度上实现“无刷新”。有关 Xajax 的内容可参看前面章节。

12.2.3 系统需求

系统包含 5 个基础的聊天用例：登录、退出、发表聊天内容、查看聊天内容以及查看在线用户列表。用例图如图 12-2-1 所示。

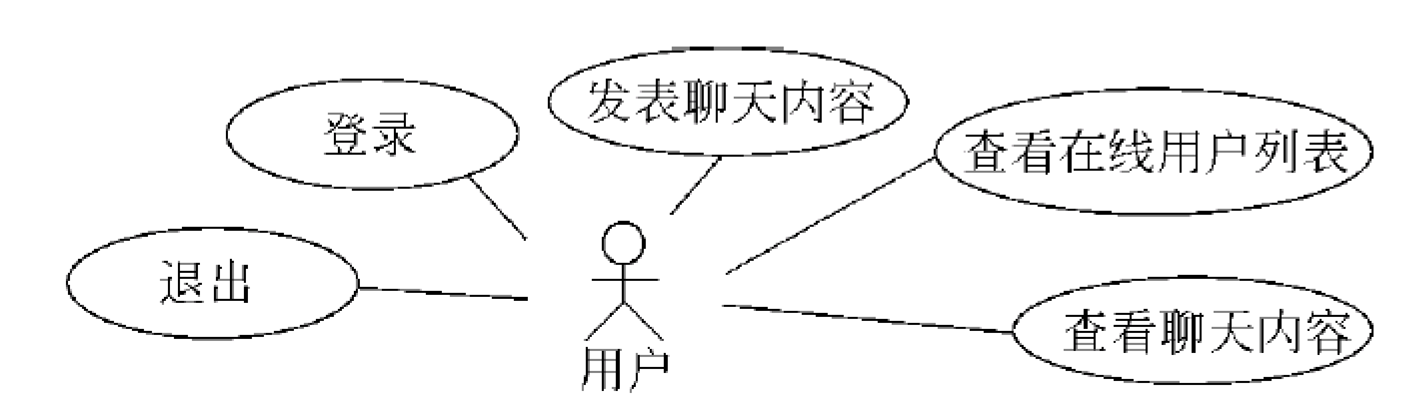


图 12-2-1 聊天用例图

登录模块让用户输入昵称和选择性别后进入聊天室进行聊天，进入聊天室后系统要记录该用户的昵称、性别和登录时间，并把该用户登记到在线用户列表，如果昵称为空或不选择性别则提示错误并拒绝用户进入聊天室。退出模块是从在线用户列表把该用户删除和退出系统。发表聊天内容模块在用户输入聊天内容、选择特定的聊天对象后发表聊天内容，聊天对象包括单个用户和全部用户，用户可以手动刷新聊天对象列表，每次发表聊天内容，系统要更新该用户的最新发表言论时间。查看聊天内容模块的功能是：每隔 2 秒客户端从服务器中读取最新的 20 条聊天记录并显示出来，聊天记录包括发表人、聊天对象、发表时间和发表内容。查看在线用户列表的功能是：每隔 10 秒客户端从服务器读取在线用户列表数据并显示出来。

12.2.4 数据库设计

系统设有两个数据库表：在线用户表和聊天内容表。在线用户表记录了所有在线用户的信息，包括用户昵称、性别、登录时间和最近发言时间等信息。聊天内容表记录所有的聊天相关信息，每条聊天信息包括发言人、聊天对象、发言内容和发表时间。在线用户表和聊天内容表具体结构如表 12-2-1 和表 12-2-2 所示。

表 12-2-1 在线用户表 userlist

字段名	字段含义	数据类型
username	用户昵称	varchar(20)
sex	性别	int(11)
loginDatetime	登录时间	datetime
lastDatetime	最近发表时间	datetime

表 12-2-2 聊天内容表 message

字段名	字段含义	数据类型
spokesman	发言人	varchar(20)
toman	聊天对象	varchar(20)
message	发言内容	text
speakDatetime	发表时间	datetime

12.2.5 数据类

系统采用面向对象编程思想，设有 user、message 和 db 类，类 user 表示在线用户，类 message 表示聊天信息，类 db 用来实现数据库操作，包括插入新的发言信息、读取发言信息、更新用户发言时间、插入新的在线用户和删除下线用户等操作。删除下线用户操作可分为删除指定用户和超时用户两种操作，当用户主动退出系统时可调用删除指定用户操作；当刷新用户列表时调用删除超时用户操作，让系统查看是否有用户太久未发言以及删除该类用户。下面是聊天室数据库类的范例。

【例 12-2-1】 （代码位置：\12\chat\db.php）

```
<?php
class user
{
    public $username;                    //昵称
    public $sex;                         //性别
    public $loginDatetime;               //登录时间
    public $lastDatetime;                //最近发言时间
}
class message
{
    public $spokesman;                   //发言人
    public $toman;                       //聊天对象
    public $message;                     //聊天内容
    public $speakDatetime;               //发言时间
}
class db
{
    public function insertMessage($msg)
    {
        //功能：插入新发言
        //$msg 是 message 类型变量
        $con=mysql_connect("localhost","root","11111111");
        if(!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("chatsys",$con);
        $spokesman=$msg->spokesman;
        $toman=$msg->toman;
        $message=$msg->message;
        $speakDatetime=$msg->speakDatetime;
        mysql_query("set names utf8");
        mysql_query("INSERT INTO message(spokesman,toman,message,speakDatetime)
```

```
    VALUES('$spokesman','$toman','$message','$speakDatetime')");
    mysql_close($con);
}

public function insertUser($user)
{
    //功能：插入新用户
    //$user 是 user 类型变量
    $con=mysql_connect("localhost","root","11111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("chatsys",$con);
    $username=$user->username;
    $sex=$user->sex;
    $loginDatetime=$user->loginDatetime;
    $lastDatetime=$user->lastDatetime;
    mysql_query("set names utf8");
    mysql_query("INSERT INTO userList(username,sex,loginDatetime,lastDatetime)
    VALUES('$username','$sex','$loginDatetime','$lastDatetime')");
    mysql_close($con);
}

public function updateUser($user)
{
    //功能：更新用户最近发言时间
    //$user 是 user 类型变量
    $con=mysql_connect("localhost","root","11111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("chatsys",$con);
    $username=$user->username;
    $sex=$user->sex;
    $loginDatetime=$user->loginDatetime;
    $lastDatetime=$user->lastDatetime;
    mysql_query("set names utf8");
    mysql_query("UPDATE userList SET lastDatetime='$lastDatetime'
    where username='$username' ");
    mysql_close($con);
}
public function getMessages()
```

```
{
    //功能：获取最近十条发言
    $con=mysql_connect("localhost","root","11111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("chatsys",$con);
    mysql_query("set names utf8");
    $result=mysql_query("SELECT * FROM message order by speakDatetime desc limit 
    10");
    $msg=array();
    $i=0;
    while($row=mysql_fetch_array($result))
    {
        $msg[$i]['spokesman']=$row['spokesman'];
        $msg[$i]['toman']=$row['toman'];
        $msg[$i]['message']=$row['message'];
        $msg[$i]['speakDatetime']=$row['speakDatetime'];
        $i++;
    }
    mysql_close($con);
    return $msg;
}

public function getUsers()
{
    //功能：获取在线用户列表
    $con=mysql_connect("localhost","root","11111111");
    if(!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("chatsys",$con);
    mysql_query("set names utf8");
    $result=mysql_query("SELECT * FROM userList ");
    $users=array();
    $i=0;
    while($row=mysql_fetch_array($result))
    {
        $users[$i]['username']=$row['username'];
        $users[$i]['sex']=$row['sex'];
        $users[$i]['loginDatetime']=$row['loginDatetime'];
        $users[$i]['lastDatetime']=$row['lastDatetime'];
```

```
            $i++;
        }
        mysql_close($con);
        return $users;
    }
    public function deleteUser($username)
    {
        //功能：从在线用户列表中删除指定用户
        $con=mysql_connect("localhost","root","11111111");
        if(!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("chatsys",$con);
        mysql_query("set names utf8");
        //从在线用户列表中删除指定用户
        $sqlcmd="delete from userList where username='$username' ";
        mysql_query($sqlcmd);
        mysql_close($con);
    }
    public function deleteUsers()
    {
        //功能：从在线用户列表中删除未发言用户
        $delUsers=array();
        $curTime=date("Y-m-d H:i:s");    //获取当前时间
        $con=mysql_connect("localhost","root","11111111");
        if(!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
        mysql_select_db("chatsys",$con);
        mysql_query("set names utf8");
        $result=mysql_query("SELECT * FROM userlist");
        while($row=mysql_fetch_array($result))
        {
            //计算最近发言至今的时间差
            $diff=strtotime($curTime)-strtotime($row['lastDatetime']);
            //记录超过10分钟不说话的用户
            if($diff>60*10)
                $delUsers[]=$row['username'];
        }
        $sqlcmd="";
        foreach($delUsers as $key=>$value)
            $sqlcmd.=" '$value',";
```

```
        //删除10分钟未发言的用户
        $sqlcmd="delete from userList where username in(".$sqlcmd."'')";
        mysql_query($sqlcmd);
        mysql_close($con);
    }
}
?>
```

12.2.6 登录模块

当用户输入昵称和选择性别成功登录后，系统使用Session把用户的昵称、性别和登录时间等信息保存起来，同时，把该用户插入到数据库在线用户表中，之后跳转到聊天主界面。如果用户昵称为空则跳出对话框提示错误。下面是登录模块的范例。

【例12-2-2】 （代码位置：\12\chat\login.php）

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>登录聊天室</title>
<script language="javascript">
window.onload=function(){
        document.getElementById('lgbtn').onclick=function(){
        name=document.getElementById('user').value;
        name=name.replace(/\s*/,'');
        if(name==''){
            alert('请输入昵称');
            form.user.focus();
            return false;
        }
    }
}
</script>
</head>
<body>
    <form>
    <div id="inputdiv">
        <div id="namediv">
          昵称:<input type="text" name="user" id="user"/>
        </div>
        性别:
        <span>
        <img src="images/male.jpg"/><input id="sex" name="sex" type="radio"
        value="1" checked="true"/>
        </span>
```

```
            <span>
            <img src="images/female.jpg"/><input id="sex" name="sex" type=
            "radio" value="0"/>
            </span>
            <input type="submit" id="lgbtn" value="进入聊天室"/>
        </div>
        </form>
    <?php
        include_once("db.php");
        session_start();
        $username="";
        $sex="";
        if(isset($_REQUEST['user'])==true)
            $username=$_REQUEST['user'];
        if(isset($_REQUEST['sex'])==true)
        {
            $sex=$_REQUEST['sex'];
        }
        //判断是否有信息输入
        if(strlen($username)>0)
        {   //登录成功则记录下用户名,性别和登录时间
            $_SESSION["username"]=$username;
            $_SESSION["sex"]=$sex;
            $_SESSION["loginDatetime"]=date("Y-m-d H:i:s");
            $_SESSION["lastDatetime"]=$_SESSION["loginDatetime"];

            $db=new db();
            $user=new user();
            $user->username=$username;
            $user->sex=$sex;
            $user->loginDatetime=date("Y-m-d h:i:s");
            $user->lastDatetime=$user->loginDatetime;
            $db->insertUser($user);            //把新用户插入到用户列表中
            header("Location:main.htm");
        }
    ?>
    </body>
    </html>
```

12.2.7 主界面框架

主界面是一个 HTML 垂直水平混合框架，该框架同时加载显示 3 个页面文件：userList.php、speak.php 和 displayMessage.php。其作用分别是显示在线用户列表、发言和显示发言信息。下面是聊天主界面框架的范例。

【例 12-2-3】 (代码位置:\12\chat\main.htm)

```
<html>
<frameset cols="10%,90%">
  <frame src="userList.php">
    <frameset rows="70%,30%">
      <frame src="displayMessage.php" frameborder="0">
      <frame src="speak.php" frameborder="0">
    </frameset>
</frameset>
</html>
```

12.2.8 刷新在线用户列表功能

在线用户列表数据存储在服务器数据库系统中,客户端需要访问服务器读用户数据才能刷新在线用户列表。系统使用 JavaScript 定时触发请求刷新界面,使用 Xajax 读取服务器中的在线用户列表数据。客户端首次运行先使用脚本 JavaScript 中的方法 setTimeout 定时 1 秒触发运行自定义的方法 refreshUserList;refreshUserList 再调用 Xajax 注册的、使用 PHP 实现的 refreshUserList 方法刷新在线用户列表,该方法对应的客户端别名是 xajax_refreshUserList;最后使用方法 setTimeout 循环定时 10 秒触发运行 refreshUserList。Xajax 注册的方法 refreshUserList 先检测和删除太久未发言的用户,再把其余的用户读出来,最后通过 Xajax 把数据传回给客户端显示出来。加大定时时间的一个好处是减轻系统压力,但实时性会打折扣;缩短定时时间会有较好的实时性,但系统压力较大。系统使用 Xajax 的好处是:让用户感觉不到屏幕闪烁;能使用 PHP 轻易地实现功能。除了提供自动刷新在线用户列表外,系统还提供了手动刷新方法。下面是在线用户列表模块的范例。

【例 12-2-4】 (代码位置:\12\chat\userList.php)

```
<?php
include 'xajax_core/xajax.inc.php';
$xajax=new xajax();
$xajax->registerFunction("refreshUserList");
function refreshUserList($arg)
{//功能:刷新在线用户列表
    include_once("db.php");
    $objResponse=new xajaxResponse();
    $text="性别 昵称<br/>";
    $db=new db();
    $db->deleteUsers();
    $users=$db->getUsers();
    for($i=0;$i<count($users);$i++)
```

```
        {
            if($users[$i]['sex'])
                $sex="images/male.jpg";
            else $sex="images/female.jpg";

            $text.="<img src='$sex'/>    ";
            $text.=$users[$i]['username'];
            $text.="<br/>";
        }
        $objResponse->assign('userList','innerHTML',$text);
        return $objResponse;
    }
    $xajax->processRequest();
    echo '<?xml version="1.0" encoding="UTF-8"?>';
    ?>
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <script type="text/javascript">
        setTimeout("refreshUserList('')",1000);
        function refreshUserList()
        {
            //功能：刷新发言列表
            xajax_refreshUserList('');
            setTimeout("refreshUserList('')",10000);
        }
        </script>
        <title>xajax example</title>
        <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
    <?php
        $xajax->printJavascript();
    ?>
    </head>
    <body>
    <div id="userList"></div>
    <button onclick="xajax_refreshUserList('');">刷新用户列表</button>
    </body>
    </html>
```

12.2.9 发言功能

下面是发言模块的范例，例中文件 speak.php 实现了发表聊天内容功能。在发言前需要检测用户是否已登录以及是否太久未发言，如果用户未登录则提示用户登录，如果超过 10 分钟没有发言则删除用户的 Session 数据以及从在线用户列表中删除该用户，强

行让用户退出系统,并要求用户重新登录才可以发言。如果用户已登录并且在发言限时范围内,就把发言信息插入到数据库并更新最近发言时间,发言信息包括发言人、聊天对象、发言内容和发言时间。发言人、聊天对象和聊天内容由 JavaScript 获取并按规定格式打包传递给 PHP 处理,其格式是:发言人:&:聊天对象:&:聊天内容。PHP 使用函数 explode 分解出发言人、聊天对象和聊天内容,发言时间由方法 date 读取服务器本地时间来获取。

【例 12-2-5】 (代码位置:\12\chat\speak.php)

```
<?php
include 'xajax_core/xajax.inc.php';
$xajax=new xajax();
$xajax->registerFunction("speak");
$xajax->registerFunction("refreshUserlist");
function speak($arg)
{   //功能:发言
    include_once("db.php");
    $objResponse=new xajaxResponse();
    session_start();
    //检测是否登录
    if(isset($_SESSION["username"])==false)
    {
        $objResponse->Alert("请登录再聊天!");
        return $objResponse;;
    }
    $curTime=date("Y-m-d H:i:s");            //获取当前时间
    $diff=strtotime($curTime)-strtotime($_SESSION["lastDatetime"]);
    //检测发言是否超时
    if($diff>60*10)
    {
        //超十分钟则强行退出系统
        $username=$_SESSION["username"];
        session_unset();
        session_destroy();
        $db=new db();
        $db->deleteUser($username);          //从在线用户列表中删除该用户

        $objResponse->Alert("久未发言,超时,请重新登录!");
        return $objResponse;
    }
    //下面的代码把发言插到数据库中
    $db=new db();
    $msg=new message();
```

```
    //分解出发言人、聊天对象和发言内容
    $messageInfo=explode(":&:",$arg);
    $msg->spokesman=$messageInfo[0];
    $msg->toman=$messageInfo[1];
    $msg->message=$messageInfo[2];
    //发言时间
    $msg->speakDatetime=date("Y-m-d H:i:s");
    //把发言插入数据库中
    $db->insertMessage($msg);
    //更新最近发言时间
    $_SESSION["lastDatetime"]=$msg->speakDatetime;
}
function refreshUserlist($arg)
{   //功能：刷新在线用户列表
    include_once("db.php");
    $objResponse=new xajaxResponse();
    $text="<select name='toman' id='toman'>";
    $db=new db();
    $users=$db->getUsers();
    for($i=0;$i<count($users);$i++)
    {
        $username=$users[$i]['username'];
        $text.="<option value='$username'>$username</option>";
    }
    $text.="</select>";
    $objResponse->assign('tomanSpan','innerHTML',$text);
    return $objResponse;
}
$xajax->processRequest();
echo '<?xml version="1.0" encoding="UTF-8"?>';
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script language="javascript">
function say(){
    //功能：对发言人、听众和发言内容进行打包
    var content=document.getElementById('content').value;
    var toman=document.getElementById('toman').value;
    var spokenman=document.getElementById('username').innerHTML;
    var text=spokenman+":&:"+toman+":&:"+content;
    xajax_speak(text);
}
function refreshUserlist(){
```

```
        //功能：刷新用户列表
        xajax_refreshUserlist('');
    }
    function exitChat(){
        //功能：刷新用户列表
        xajax_exitChat('');
    }
    </script>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
    <title>发言</title>
    <?php
        $xajax->printJavascript();
    ?>
    </head>
    <body>
    <?php
    include_once("db.php");
    session_start();
    ?>
        <form action="exit.php" target="_parent">
        <div id="speakdiv">
            <div id="namediv">
                <?php
                    //判断是否登录
                    if(isset($_SESSION["username"])==false)
                    exit();
                    $username=$_SESSION['username'];
                    echo "<span id='username'>$username</span>";
                ?>
                向
                <span id="tomanSpan">
                    <select name="toman" id="toman">
                    <?php
                      //获取在线用户列表并显示出来
                      $db=new db();
                      $users=$db->getUsers();
                      for($i=0;$i<count($users);$i++)
                      {
                          $username=$users[$i]['username'];
                          echo "<option value='$username'>$username</option>";
                      }
                    ?>
                    </select>
```

```
            </span>
            <button onclick="refreshUserlist();">刷新</button>
            说:
            <input type="text" name="content" id="content" size="50"/>
            <button onclick="say();">发言</button>
            <input type="submit" value="退出系统"/>
        </div>
    </div>
    </form>
</body>
</html>
```

12.2.10 显示聊天内容功能

显示聊天内容和刷新在线用户列表的实现原理是相同的,JavaScript 负责定时请求刷新显示聊天内容,Xajax 负责从服务器上读取聊天内容并显示在客户端浏览器上。每隔 2 秒钟,JavaScript 就会发出刷新请求调用 Xajax 注册的函数 refreshMessageContent,该函数在 JavaScript 中对应的名称是 xajax_refreshMessageContent。refreshMessageContent 先通过类 db 中的方法 getMessages 读取最新的 10 条聊天记录,然后把这 10 条聊天记录格式化,最后把格式化后的聊天内容显示在 id 值为 MessageConent 的 DIV 块中。下面是聊天内容模块的范例。

【例 12-2-6】 (代码位置:\12\chat\displayMessage.php)

```
<?php
include 'xajax_core/xajax.inc.php';
$xajax=new xajax();
$xajax->registerFunction("refreshMessageContent");
function refreshMessageContent($arg)
{ //功能:刷新聊天内容
    include_once("db.php");
    $db=new db();
    $msgs=$db->getMessages();
    $text="\n";
    for($i=0;$i<count($msgs);$i++)
    {
        $text.=$msgs[$i]['speakDatetime']." ".$msgs[$i]['spokesman'];
        $text.=" 对 ".$msgs[$i]['toman']."说: ".$msgs[$i]['message']."\n";
    }
    $objResponse=new xajaxResponse();
    $objResponse->assign('MessageConent','value',$text);;
    return $objResponse;
}
```

```
$xajax->processRequest();
echo '<?xml version="1.0" encoding="UTF-8"?>';
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>登录聊天室</title>
<script type="text/javascript">
    setTimeout("refreshMessageList('')",2000);
    function refreshMessageList()
    {
        //功能:刷新发言列表
        xajax_refreshMessageContent('');
        setTimeout("refreshMessageList('')",2000);
    }
</script>
<?php
    $xajax->printJavascript();
?>
</head>
<body>
<textarea id="MessageConent" rows="20" cols="100" style="overflow-x:hidden;overflow-
y:scroll;">
</textarea>
</body>
</html>
```

12.2.11 退出系统功能

当用户太久未发言，或者单击发言界面中的“退出系统”按钮时，用户将退出系统。系统如何知道用户太久未发言呢？当用户单击“发言”按钮要发言时，系统就会检测用户是否太久未发言，如果久未发言则强行退出系统，这个功能已在发言模块中实现，请参看12.2.9节的内容。当用户单击“退出系统”按钮时，客户端会提交表单并跳转到页面exit.php，页面exit.php会先把用户的Session数据清除干净，然后再从在线用户列表中删除该用户，最后跳转到登录页面。在发言功能中，通过表单中的action="exit.php"实现了客户端提交表单并跳转到页面exit.php，通过target="_parent"使跳转页面显示在整个浏览器中，而不是主界面中的子框架。清除用户Session数据、从在线用户列表删除用户和页面跳转的实现过程如下例所示。

【例12-2-7】 (代码位置：\12\chat\exit.php)

```
<?php
```

```
    include_once("db.php");
    session_start();
    $db=new db();
    $username=$_SESSION["username"];
    $db->deleteUser($username);            //从在线用户列表中删除该用户
    session_unset();
    session_destroy();
    header("location:login.php");
?>
```

12.2.12 运行系统

为方便描述，假设服务器 IP 为 127.0.0.1。先把聊天室系统部署在服务器上，然后输入 http://127.0.0.1/chatSys/login.php 进行登录，登录界面如图 12-2-2 所示。

如果昵称为空，那么单击“进入聊天室”按钮将弹出“请输入昵称”警告对话框，运行结果如图 12-2-3 所示。

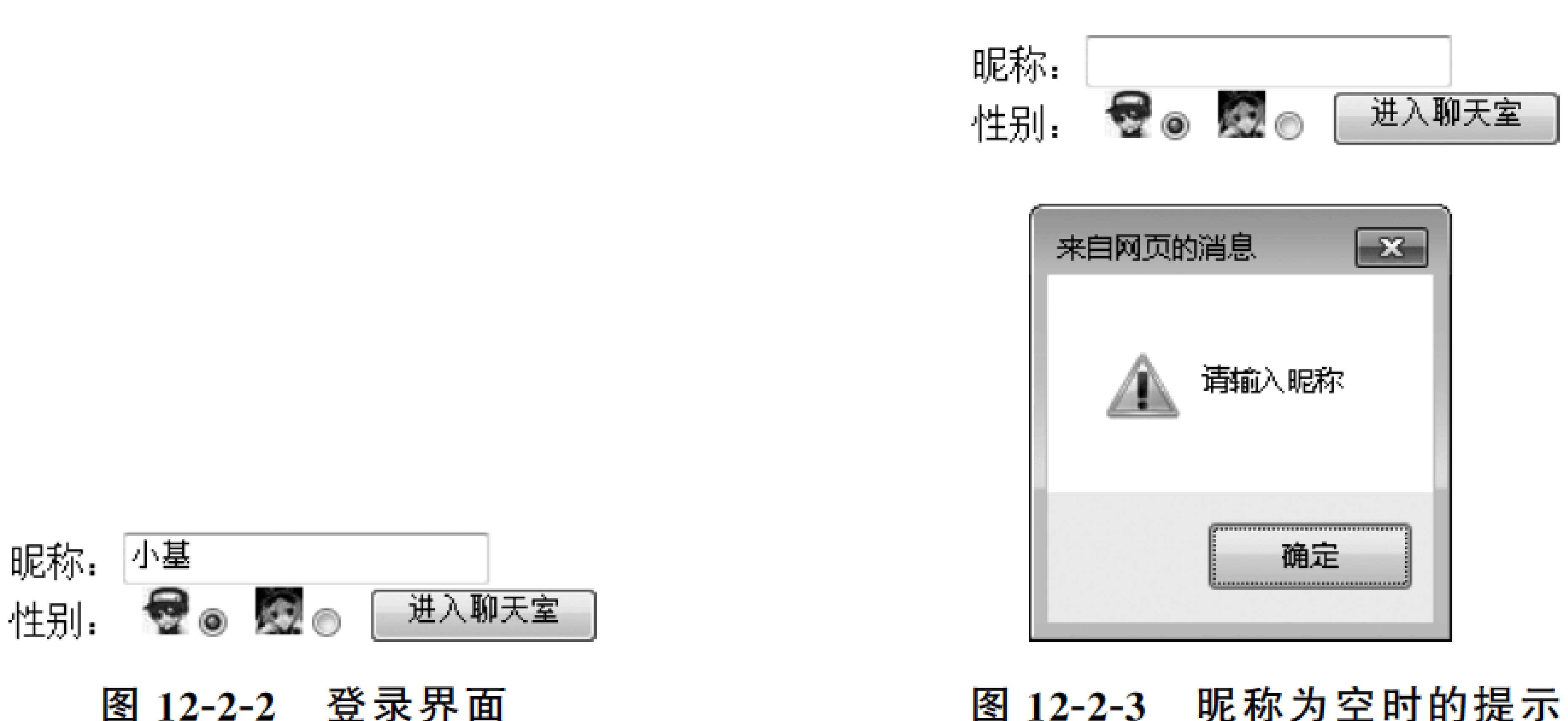


图 12-2-2　登录界面　　图 12-2-3　昵称为空时的提示

当输入昵称和选择性别后，单击“进入聊天室”按钮将进入聊天主界面，如图 12-2-4 所示，图最左边是在线用户列表，图右边的上半部分是显示聊天内容区域，下半部分是发言区域。

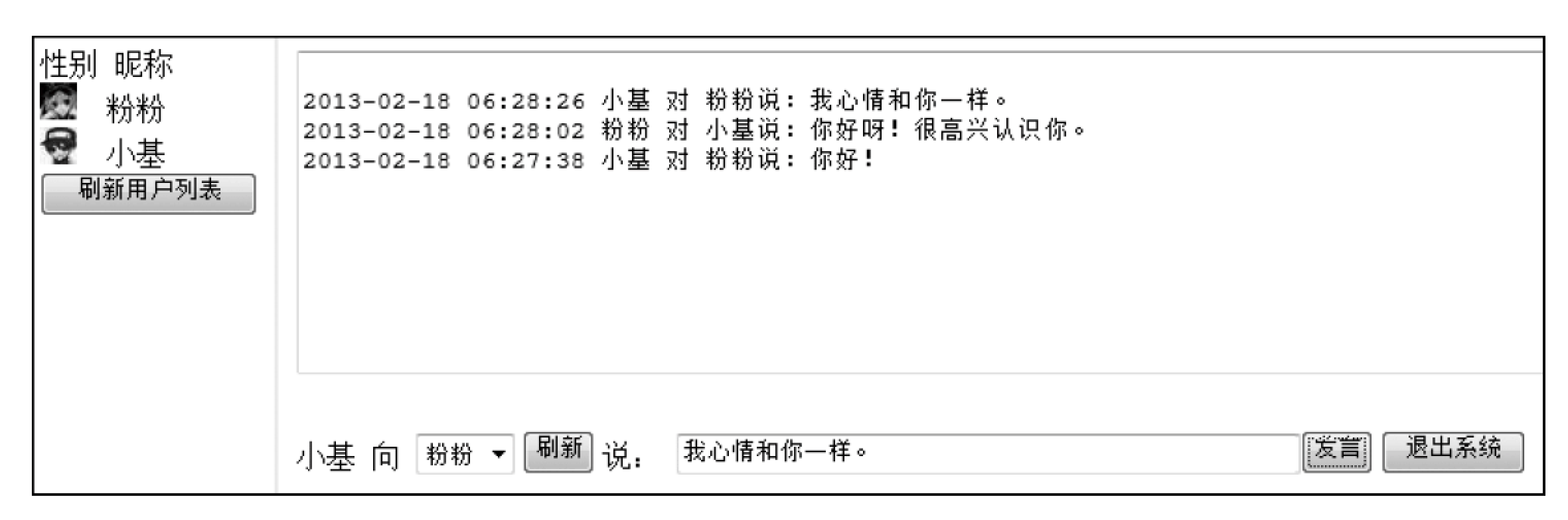


图 12-2-4　聊天主界面

用户可以在发言区域选择聊天对象和输入聊天内容后单击“发言”按钮进行聊天。

如果在线用户列表中的用户没有显示在聊天区域中,可以单击“刷新”按钮更新用户。如果要实时查看在线用户列表,可以单击其下的“刷新用户列表”按钮更新列表内容。如果想退出系统可单击“退出系统”按钮。如果用户单击“发言”按钮且之前超 10 分钟未曾发言,则弹出对话框提示用户发言超时,如图 12-2-5 所示。



图 12-2-5　发言超时

第 13 章 chapter 13

WAP 开发

13.1 WAP 概述

13.1.1 WAP 概念

手机上网无疑是如今最热门的一项技术。与传统的 PC 相比，手机受到 CPU、内存、电池寿命和屏幕等更多因素的限制，而且无线网络与有线网络相比一般具有低带宽、高延迟和稳定性差等缺点，这给手机上网带来了许多技术上的难题。本章主要讨论非智能手机上网技术，而非讨论智能手机上网技术。WAP（Wireless Application Protocol）中文翻译为无线应用协议，它是在数字移动电话、Internet 或其他个人数字助理机（PDA）、计算机应用之间进行通信的开放全球标准。简单点说，它规范化了移动设备端的网页浏览格式及通信协议。WAP 的本质就是 Internet，而且是可以移动的 Internet。WAP 的诞生是 WAP 论坛成员努力的结果，WAP 论坛是在 1997 年 6 月由诺基亚、爱立信、摩托罗拉和无线星球（Unwired Planet）共同组成的，该论坛的网址是 www.wapforum.com。WAP 2.0 是目前较新的版本，WAP 的目标是通过 WAP 将 Internet 的大量信息及各种各样的业务引入到移动电话等移动终端之中。无论在何地何时，只要需要信息或处理业务，就可以打开 WAP 手机，享受无穷无尽的网上资源或移动办理业务，如综合新闻、天气预报、股市动态、商业报道、当前汇率和电子银行等。

GPRS 是 General Packet Radio Service 的缩写，中文可译为通用分组无线业务。GPRS 是一种高速、高效的无线系统，可以通过 GPRS 手机发送及接收电子邮件，还可访问 Internet 和下载资料，也可以通过“蓝牙”等设备在计算机上利用 GPRS 无线上网。GPRS 具有“高速”和“永远在线”的优点。在性能方面，GPRS 比电路交互数据业务速度高将近 4 倍，最高速率达 115kb/s。在“永远在线”方面，用户访问互联网时，手机就在无线信道上发送和接收数据，就算没有数据传送，手机也一直与网络保持连接，不像普通拨号上网那样断线后还得重新拨号才能上网冲浪。GPRS 与 WAP 属于不同的范畴，它不会取代 WAP，它的目的是将 Internet 的丰富信息及先进的业务引入到移动电话等无线终端之中。作一个比喻，GPRS 是公路，而 WAP 是在公路上的汽车，WAP 原先是运行在 2 车道的 GSM 公路上，而现在是运行在 8 车道的 GPRS 公路上，GPRS 公路上不仅可以跑普通汽车，还可以开赛车等，其业务非常广阔。可见，WAP 是高层应用，而 GPRS 是底

层传输。引用邮递信件的例子来说明 WAP 与 GPRS 的关系：如果说 WAP 是平信、贺卡或明信片，那么 GPRS 就是运送信件的交通工具。中国移动开通 GPRS 之后，WAP 就行驶在 GPRS 公路上，而行驶在 GPRS 的公路上可以提高数据传输速度。一般 GPRS 分包月（每月固定流量）和按流量计算费用，中国移动手机上附有的 GPRS 一般不用专门设置，如果手机屏幕上有 GPRS，说明已开通该服务，从一开机 GPRS 就附在网络上（睡觉）不收费，当手机应用程序开始使用 GPRS 传输数据时，它才从“睡觉”中醒来“真正”接通网络。手机已走过了 1G（第一代）、2G（第二代）和 3G（第三代）手机通信技术，目前正要进入第四代通信技术，GPRS 是介于 2G 和 3G 间的手机通信技术，它用于 2G 向 3G 过渡。比 GPRS 具有更高带宽的第三代网络仍需要 WAP 吗？就算是带宽增加了，带宽的使用不会变成零成本，其因是终端的电力消耗更大，射频方面的成本更高，射频频谱资源的利用程度更高和网络负荷增加等。原来的限制因素在第三代仍然存在，如间断的信号覆盖、细小的屏幕、低功率的设备、各类不同的数据模式和制式等，说明有必要继续优化无线环境中的设备和系统资源。

WAP 的发展与应用是无可限量的，可以说唯一的限制就是用户的想象力，WAP 不但使现有的许多应用得到了突飞猛进的改变，同时也催生出更多崭新的增值业务。比如用来支持特定商业程序、信息发送或领域维护，其中包括客户服务与备件提供、消息通知与呼叫管理、电子邮件、基于电话的增值业务、群体计划、气象与交通信息、地图与位置服务、新闻与体育报道等；尤其值得注意的是，它使得广泛应用于网上的信息服务，逐渐由纯信息的提供向更加交互化与最终电子商务化发展。

13.1.2 WAP 与 WWW

WAP 的发展离不开 WWW，实际上，整个 WAP 协议的制定参照了 WWW 的模型，并且尽可能地沿用了 Internet 的诸多标准和技术，如 WAP 中的 WML 语言严格遵守 XML 语法，其作用和 HTML 类似，用于描述静态页面，WMLScript 与 JavaScript 等脚本语言也很相似，用于描述动态页面。用户使用支持 WAP 的手机上网，网页的静态内容用 WML 语言描述，其动态内容用 WMLScript 语言描述，或者说，WAP 移动终端上的浏览器所识别的描述语言不再是熟悉的 HTML 或 JavaScript，而是 WML 和 WMLScript 语言。尽管 WAP 参照 WWW 模型，但它们也有很大的区别。简单地说，计算机上打开的网页都是 WWW 的，非智能手机上打开的是 WAP 的，但非智能手机上如安装有第三方的浏览器软件，也可以打开计算机上能打开的 WWW 网页，不过某些动态网页解释语言的能力很差。技术方面，WAP 中的 WML 相对 WWW 用的 HTML 语言更简洁，可以节省流量和加快访问速度。

在 WAP 协议中提到 WAP 页面所使用的标记语言为 WML，当需要进行较高级的操作，如使用终端机的某些资源时，可以使用 WMLScript 脚本语言，这两种语言很简单，甚至可以用简陋来形容，特别是它们对数据库的操作几乎无能为力，这种“缺陷”与 WAP 广阔的商务和应用前景格格不入，但并这并非无能为力，由于 WAP 和 WWW 的交融，仍然可以使用 WWW 的技术与资源来解决数据库的访问问题，因为毕竟多数数据库仍处于有线网络内。

13.1.3 WAP工作原理

WAP网络架构由3部分组成，如图13-1-1所示，即WAP网关、WAP终端和WAP服务器，三者缺一不可。其中。WAP网关主要作用是联系GPRS网与Internet的桥梁；WAP终端为用户提供了上网浏览器等；WAP内容服务器可以存储大量信息，以供手机访问、浏览和查询等。

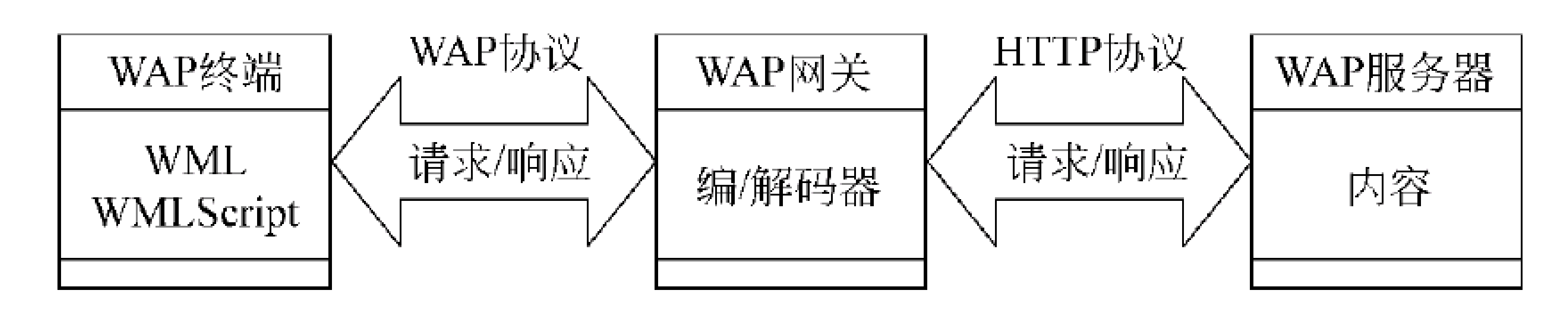


图13-1-1 WAP系统框架

WAP终端通过浏览器首先与WAP网关建立连接，用WAP协议向WAP网关发出获取信息请求，请求消息中带有标识所要获取的文档的URL。请求信息是紧缩的二进制码形式，按照WAP协议栈的消息格式装在数据包中。WAP网关收到WAP消息格式的数据包后，拆开数据包，取出请求消息。然后将二进制码形式的请求消息解码，将其转换成ASCII码，再按照HTTP协议的格式封装。将HTTP消息封装在TCP/IP的数据包中，发送到Internet上。WAP服务器收到WAP网关的请求消息后，作相应处理，发出包含信息内容的响应，通过Internet送到WAP网关。在WAP服务器看来，WAP网关相当于WWW客户机，WAP网关收到的响应是通过TCP/IP协议传送过来，并且信息仍然是ASCII码形式。WAP网关从TCP/IP数据包中取出信息，然后把ASCII码形式的信息内容转换成紧缩的二进制码。接着，再把二进制码的信息内容按照WAP规定的消息格式封装，通过WAP协议发送给WAP终端，WAP终端收到响应，解数据包，取信息内容，用户即可浏览所要获取的信息。WAP网关可以理解成两个功能：协议转换功能以及内容编码和解码功能。协议转换功能的作用是将来自WAP协议栈的请求翻译到WWW协议栈，或将来自WWW协议栈的响应翻译到WAP协议栈。内容编码和解码功能主要是将Web内容翻译成紧密编码的格式，以减少通过无线数据网络传输的数据包的大小和数量。

13.1.4 WML和WMLScript简述

WML(Wireless Markup Language)中文名为无线标记语言，它是一种基于XML的标记语言，和HTML一样，WML的作用是用于显示数据。内置于移动设备中的微型浏览器能够解释这种标记语言。虽然它和HTML语言很相像，但WML是基于XML的，更准确地说WML是XML的一个应用，在语法上WML比HTML更加严格。用WML编写的页面叫DECKS。HTML页面可以用IE等浏览器解释和显示，而WML语言编写的DECKS是专门用在手机等无线终端显示屏上显示供人们阅读的，并且同样也可以向使用者提供人机交互界面，接受使用者输入的查询等信息，然后向使用者返回他所想要获取的最终信息。

为什么在手机等无线终端上使用WML而不是使用广泛的HTML？一个原因是WAP协议要求使用WML。另一个原因是WML与HTML相比需要很小的带宽。随着无线技术的提高，带宽会越来越宽。这个原因可能不再是主要原因，但是在达到高带宽之前需要许多年的时间。另外一个更重要的原因是HTML比WML需要计算能力更强的处理器。处理器的能力强意味着需要更强的能量。在无线设备中能量来自电池，移动设备中的电能是有限的，而一个低处理能力的处理器就意味着电池的使用时间可以更长。除此，HTML需要大尺寸的显示屏而WML只需要小尺寸的显示屏也是重要的原因。

HTML格式是否能转换成WML格式？从技术上讲是可以进行转换的，有特定的系统可以实现HTML格式自动转换为WML格式，而这一转换的实现需依设备、应用程序以及HTML页面格式的复杂性而定。大多数WAP移动电话诸如Nokia7110只有大小十分有限的显示屏及输入装置，若要浏览信息，只有从技术上着手。大多数HTML页面格式并不支持WAP的有效工作架构，而对于用户而言，这将使得自动转换成为一个既耗人耐性又耗人费用的过程。

WMLScript是一种在手机端运行的，经过优化，更利于WAP应用的脚本语言，它是在WML页面中使用的脚本语言，是一种轻量级的JavaScript语言，尽管WMLScript与JavaScript相似，但在功能上没有JavaScript丰富。不像将JavaScript嵌入到HTML中那样，WMLScript必须包含在一个独立的文件中，WMLScript并不嵌在WML页面中，WML页面仅仅包含对脚本URL的引用。WML脚本在WAP浏览器运行之前，需要先在服务器上编译为字节编码。WMLScript的设计宗旨是为WMLScript系统提供一般的脚本处理能力，使用WMLScript可以进一步扩展基于XML的WML语言的编程功能，开发针对窄带宽的网络应用及内容，如文本、图像和选择列表等，可以使用简单的格式编写出更灵活和更具可读性的用户界面。WMLScript具备WML所不能具备的主要优点和功能包括如下几个方面：

(1) 可检查用户输入的合法性。

(2) 扩展用户浏览器的功能，比如允许程序员开发手机的电话呼叫、发送短信息、存储电话号码、管理电话簿或SIM卡等。

(3) 生成用户端的确认、提示、警告信息或操作对话框，并使之快速显示在浏览器上。

(4) 在用户浏览器有了更改后，能够对浏览器端的软件和参数进行扩展与配置。

(5) 最大程度克服客户端的窄带宽通信连接限制，并提供丰富的程序功能。

(6) 补充WML并使之实现针对微型移动终端设备的多种服务，如支持高级用户界面、增加客户端智能型、提供用户浏览器外围功能的访问能力，以及在服务器与客户端浏览传输数据时减少带宽占用等。

与WML一样，WMLScript也沿用了WWW和HTML访问资源的URL、HTTP等规范，并扩大了URL使用的范围。在WMLScript中，不仅超链接、文件路径可以作为URL处理，外部函数、访问控制信息等也可以作为URL处理。

13.2 WAP开发环境搭建

为了支持 WAP 开发，需要让服务器和客户端浏览器都支持 WAP。在 WAP 编程与开发中，为了对所编写的 WAP 应用进行测试，通常要建立 WAP 的测试环境。一般来说，WAP 测试环境可以从浏览器环境、模拟环境和实际环境 3 个方面建立。

浏览环境的建立十分简单。目前 Internet 尚有许多站点提供 WML 浏览器的免费下载服务。比较著名的 WML 浏览器是 Winwap(论坛中提供下载)。各移动通信设备公司均提供各自的浏览器。安装这些浏览器后，用户就可以在 Windows 系统环境下访问 WAP 站点，查看 WAP 页面。

用于 WAP 测试的模拟环境是通过使用移动通信设备公司所提供的 WAP 手机模拟器来实现 WML 浏览的。虽然说这种模拟环境提供了集成环境及与 WAP 手机基本一致的模拟器，但仍难保证所用模拟器与其实际产品完全一致，尤其是没有 WAP 网关的参与，因此这是一种并不完备的检测。特别是在这种模拟环境下与无线电话应用(Wireless Telephony Application，WTA)相关的服务根本没有办法进行检测。不过，对于单纯的开发测试来说，这样模拟环境基本能满足要求。

WAP 测试的实际环境中需要 WAP 手机、网卡及服务器 3 个部分，因此，为了建立 WAP 测试的实际环境，开发者需要购买一些主流的 WAP 手机，同时使用前面介绍的方法在原来的 HTML 服务器上建立一个 WAP 专用的虚拟目录来创建 WAP 服务器，然后使用现有网关或移动通信公司提供的相应网关。只要这三者都能顺利正确地工作，开发者就可以通过 WAP 手机对 WAP 网页及应用进行测试了。

下面介绍如何让服务器和客户端支持 WAP。

(1) 让 Apache Web 服务器支持 WAP。修改 Apache 的 httpd.conf 文件，增加如下内容：

```
AddType text/vnd.wap.wml.wml
AddType application/vnd.wap.wmlc.wmlc
AddType application/vnd.wap.wmlscriptc.wmlsc
AddType image/vnd.wap.wbmp.wbmp
```

重新启动 Apache Web 服务器后，该服务器就能识别 WML 文件并支持 WAP 开发。

(2) 客户端支持 WAP。Opera 不是一款开源的浏览器，但它是免费使用的。通过它，可以在 PC 上浏览 WAP 网站，对开发人员来说，这会使开发工作简洁方便许多。当然，Opera 浏览器也可以安装在各种操作上运行，包括手机上的操作系统。可以从官方网站上下载并安装该浏览器，其官方网站是 http://www.opera.com/download。当安装该浏览器后，客户端就已经支持 WAP 了。

13.3 编写和运行 WML 程序

WML 文件可以使用记事本等工具进行编辑。EasyPad WAPtor V2.3 是一款简单易用的 WAP 页面编辑工具，特别适合刚开始学习编写 WAP 网页的人使用。除此，还可

以借用 Ericsson WapIDE 和 Nokia WAP Toolkit 工具开发 WAP 程序。建立 WAP 站点比起 Web 站点来要简单得多，这意味着不需要花更多时间来学习它。可从一些大型网站下载 WAPtor 软件，该软件的使用较简单，可通过菜单中的 File→New 命令新建一个 WML 文件，该软件自动会建立一个 WML 模板文件，其内容如以下范例所示。

【例 13-3-1】 （代码位置：\13\testWML. wml）

```
<?xml version="1.0"?>
<!--created by WAPtor(http://www.waptop.net/)-->
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml"
>
<wml>
<!--THIS IS THE FIRST CARD IN THE DECK-->
<card id="MainCard" title="This is a first card">
<p align="center">
  This is a sample text
</p>
</card>
</wml>
```

输入上述代码后，大家一定想查看一下它的运行效果。那么使用什么工具查看运行结果呢？最直接的工具是 WAP 手机，但这需要 WAP 服务供应商的支持，对大多数开发人员来说不一定能实现。较方便的方式是用微浏览器，例如 Opera 浏览器，它能在 PC 桌面运行直接解释和显示 WML。先把上述代码保存成文件 testWML. wml 并复制到 PHPNow 套件解压目录下的 htdocs 子目录，即 C:\PHP\htdocs，这项工作实际是把程序部署到了服务器以便客户端能访问程序。然后打开 Opera 浏览器并输入地址 http://127.0.0.1/testWML. wml，将会看到运行结果如图 13-3-1 所示。

This is a first card
This is a sample text

图 13-3-1　testWML 运行结果

13.4　WML 语言

13.4.1　语法结构分析

下面是一个普通的 WML 页面，该页面有两张卡片，每张卡片都包含有测试用的文字。

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.
1.xml">
```

```
<wml>
  <card id="card1" title="Card 1">
    <p>
Hello World!
This is the first card.
</p>
  </card>
  <card id="card2" title="Card 2">
    <p>
This is the second card.
Good Bye.
</p>
  </card>
</wml>
```

下面对 WML 页面语法和格式进行分析。

(1) 一个有效的 WML 页面也就是一个有效的 XML 文件，因此必须包括一个 XML 声明以及一个文件类型声明。一个 XML 声明总是在文件的第一行，注意前面最好不要有空格或者还行，其声明格式如下：

```
<?xml version="1.0"?>
```

紧跟着的是 DOCTYPE 声明，注意声明字母的大小写不要搞错，其文件类型声明如下：

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
```

(2) 标签。在 WML 语言中需要使用标签(tag)，其使用形式与 XML 中的形式是完全一致的。例中的<p>、</p>、<card>和</card>等皆为标签。

(3) 元素。WML 元素(element)用于描述卡片组(deck)的标记信息及结构信息。一个元素通常由一个首标签、内容、其他元素及一个尾标签组成，具有下述两种结构之一：

```
<首标签>内容 </尾标签>
```

或

```
<标签/>
```

元素包含的内容中还可以有元素，这些元素也是由首标签、相应内容、其他元素及尾标签组成。不包含内容的元素称为空元素，它退化成一个单独的标签，或者说，单独的标签也是一个元素。

(4) 属性。WML 与 XML 一样，其标签可以包含很多属性。属性用于给标签提供必要的附加信息，且属性内容通常在起始标签内使用。不过，属性内容不会被浏览器显示，它只作为参数为标签提供必要的信息。指明属性值的时候，需要把该值用引号括起来，可以是单引号或者双引号，引号通常成对嵌套使用。属性名称必须小写。例如：<card

id="card1" ontimer="#card2" title="demo">。而且，单引号的属性中还可以包含双引号的属性。实体字符也可作为属性值。实体字符是指诸如&、<、>、'、"等特殊字符，在WML程序中显示这类字符需要特殊处理。

(5) 文档结构。WML文档是由“卡片”(card)和“卡片组”(deck)构成的，一个deck是一个或多个card的集合。当客户终端发出请求之后，WML即从网络上把deck发送到客户的浏览器，这时用户就可以浏览deck内包含的所有card，而不必从网上单独下载每一个card。程序中的第一个card是默认可见的card。文档的实体包含在<wml>…</wml>标记中，文档里每个card又包含在<card>…</card>标记中，实际的文字段落则包含在<p>…</p>标记对中。

注意，deck是一副纸牌的意思，这里是指一叠卡片，可以把它翻译为“卡片组”。另外，card指的是WAP手机屏幕大小的网页，尽管有时一个card可能需要多屏才能显示完，但也可以把它翻译为“页面”，不过这样与HTML中的“页面”不易区分，所以还是采用“卡片”的译法较好。

(6) card标签。一个deck可以包含多个card，每个card的内容可能不止一屏显示。对于每一个card，WML均使用<card>和</card>进行包含和定义。<card>同时可以包含文本、标记、链接、输入控制、任务和图像等内容。卡片之间的导航是通过客户端(移动电话)完成的，不需要对服务器的额外访问。

(7) 大小写敏感。无论是标签元素还是属性内容都是大小写敏感的，这一点继承了XML的严格特性，任何大小写错误都可能导致访问错误，这是WML制作者必须注意的问题。

13.4.2 card元素

card标签定义了一个卡片的各种属性和包含内容。它的语法格式如下：

```
<card id="name" title="label" newcontext="boolean" ordered="true" onenterforward="
href" onenterbackward="href" ontimer="href">
内容
</card>
```

card元素属性的功能及用法如下：

(1) id。用于指定card的名字。

(2) title。用于为卡片指定一个简单的标题或说明信息，一般显示在屏幕上方。

(3) newcontext。用于指示用户重新进入本卡片的时候是否需要初始化卡片中所有的内容，即是否清除以前保留的信息。它有true和false两种值，true值表示要初始化卡片，否则将不进行初始化设置，也不清除历史记录。其默认值为false。另外，newcontext仅当作为go任务的一部分时才可被执行。

(4) ordered。它有true和false两种布尔值，默认值是true。它用来表明该卡片里的内容是按固定的顺序显示，还是按用户的选择顺序来显示。默认的是按线性顺序显示，即按代码的顺序。为了方便填表单，当ordered设置为true时，如果一个表单的内容

不能在一屏里显示完，就分成多屏显示；当 ordered 设置为 false 时，手机可以显示一个概要卡片来总结有效的选项，用户可以从中选取表单选项来填写。但是，要注意的是，以下3个标签必须按顺序来写：<onevent>、<timer>、<do>（这和以后要讲的“事件”有关）。

（5）onenterforward。onenterforward 事件仅当用户使用 go 任务或类似于 go 的任务位和浏览卡片时才可发生，即如果用户执行 go 任务，则浏览器就会定位<go>标签中指定超链（href）的 URL 指定的卡片。card 元素中的 onenterforward 属性是 onevent 元素的一个简单格式，用于直接指定 onenterforward 事件的 URL 地址。

（6）onenterbackward。该属性可以指定其响应时间的 URL 地址。如果用户浏览时执行 prev 任务，那么浏览器就会定位到该属性所指定超链（href）的 URL 地址，并打开 URL 指定的卡片。onenterbackward 属性也属于 onevent 元素的一个简单格式。

（7）ontimer。当指定时间 timer 过期的时候，浏览器根据 ontimer 属性指定的 URL 打开相应的卡片。

下面是关于 card 元素的范例，该范例包含有3张卡片，系统会先显示第一张卡片，过了指定的时间后，会自动显示第二张卡片，再过了指定时间，会自动显示第三张卡片。

【例 13-4-1】（代码位置：\13\testCard.wml）

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" ontimer="#card2" title="OnTimerDemo">
<timer value="50"/>
This is the first card.
</card>
<card id="card2" ontimer="#card3" title="OnTimerDemo">
<timer value="50"/>
This is the second card.
</card>
<card id="card3" title="OnTimerDemo">
This is the third card.
</card>
</wml>
```

程序中<timer>标签用来定义一个计时器，其属性 value 表示计时时间，单位是1/10秒，值50表示计时5秒。当计时器计时结束后，会产生 ontimer 事件，卡片可以在 ontimer 事件中跳转到对应的卡片进行处理。该程序运行后将在 WAP 浏览器依次显示3屏信息。先显示“This is the first card.”，然后显示“This is the second card.”，最后显示“This is the third card.”。显示时每屏都有标题 OnTimerDemo，相邻两屏之间延时5秒。

13.4.3 wml和head元素

<wml>标签和HTML中的<html>标签一样，用来表明这是一个WML的deck。和HTML一样，<head>标签包含了该deck的相关信息。<head>标签之间可以包含一个<access>标签和多个<meta>标签。

13.4.4 access元素

<access>标签定义关于WML卡片组的访问控制的信息。如果<access>标签没有设置，任何其他deck/card都能够访问此deck。<access>标签永远位于<head>标签内部。每个deck只能有一个<access>标签。该标签格式如下：

```
<access domain="域" path="/路径"/>
```

属性domain和path表示能够访问此deck中card的其他deck的URL。domain用来指定域，默认值为当前域，path用来指定路径，默认值为"/"，即根目录。这两个属性是可选的。

下面范例中的deck访问权限受限于域mydomain.com，即只有来自mydomain.com的卡片才能访问该deck。

【例13-4-2】（代码位置：\13\testAccess.wml）

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
<access domain="mydomain.com"/>
</head>
<card id="card1" title="Demo">
Access Demo.
</card>
</wml>
```

13.4.5 meta元素

meta元素用于提供当前的deck的元信息，包括内存数据处理方式、数据传输方式和处理方式，针对搜索引擎对关键字和页面内容进行描述，以及设置刷新频率等。该元素是由一个单独的标签即<meta/>标签实现的元素，其语法格式如下：

```
<meta name="name"|http-equiv="name" content="value" forua="true|false"/>
```

其中，name属性和http-equiv属性只能选择使用一个；content属性是必选的，其值根据属性而定；forua属性为可选属性。各属性功能及用法说明如下所示。

(1) content。该属性用于描述 name 属性的具体内容。

(2) name。用于在页面上为描述的事物规定一个名称。可以是 keywords 和 author 等。

(3) http-equiv。该属性用于替代 name 属性,可将 meta 数据转为 WSP 或 HTTP 协议的响应头。

(4) forua。该属性用于指定那些开发者希望传送至用户浏览器的性质。它有 ture 和 fales 两个取值,如果取 false,则卡片组在发送往客户端以前必须用中间代理去除 meta 元素信息,这是因为传输的协议可能改变;若取值为 true,则 meta 数据必须如实送往用户的浏览器。默认的状态下,该属性的值为 false。

下面是 meta 元素的范例。

```
<meta http-equiv="Cache-Control" content="max-age=3600"/>
```

这条语句指定 deck 在手机内存缓存中的存储时间段,默认为 30 天(除非内存耗尽),在该期间,手机对于访问过的 deck 直接从缓存里调用。如果信息是对时间敏感的,可以用 max-age 指定 deck 在缓存里的生存期,最小单位是秒,如果指定为 0,则每次都需通过连接服务器来调用该 deck。

针对搜索引擎定义关键字:

```
<meta name="keywords" content="HTML,DHTML,CSS,XML,XHTML,JavaScript,VBScript"/>
```

定义页面内容的一些描述:

```
<meta name="description" content="Free Web tutorials on HTML,CSS,XML,and XHTML"/>
```

定义页面最近一次修改:

```
<meta name="revised" content="Hege Refsnes,6/10/99"/>
```

每 5 秒刷新一次页面:

```
<meta http-equiv="refresh" content="5"/>
```

13.4.6 文本元素

WML 程序中,为使显示的文本呈现出丰富的样式,WML 提供了一些用于格式化的元素,通过这些元素及其相应的标签可以对文本进行标注和控制,从而实现不同的显示效果。WML 提供了<p>、
和<table>等元素,分别用于分段、换行和定义表格。

br 即 break,是用于换行的元素,它是使用单独的
标签进行定义的。br 元素的作用相当于插入一个回车符。p 即 paragraph,p 元素用于划分段落,使当前文本换行并插入一个空白行。p 元素可以使用单独的<p/>标签进行定义,也可以使用<p>和</p>标签成对地进行定义,其语法格式如下:

```
<p id="id" align="alignment" mode="wrapmode"/>
```

或

```
<p id="id" align="alignment" mode="wrapmode"/>
内容
</p>
```

下面是p元素中的属性说明。

(1) align。该属性用于设置段落在浏览器中的对齐方式,有left、center和right三种取值。这三种参数值分别表示p元素当前定义的文本段落与浏览器窗口的左侧、中间和右侧进行对齐。默认值为left,即段落与浏览器窗口的左侧对齐排列。

(2) mode。该属性用于指定段落是否换行,其值为wrap表示换行,为nowrap表示不换行。

(3) id。为此元素设置的唯一名称。

WML中的表格是按照行、列进行组织的。一个表格由若干行组成,每行由若干列组成。WML提供table、tr和td元素分别用来定义表格、表格行和单元格。td元素的语法格式如下:

```
<td>单元格内容</td>
```

tr元素的语法格式如下:

```
<tr>
<td>单元格内容</td>
</tr>
```

table元素与tr元素、td元素一起,可用来创建能容纳文本和图像的表格,并可设置表格各列中文本和图像的对齐方式。其语法格式如下:

```
<table align="alignment" title="label" columns="n">
```

或

```
<table align="alignment" title="label" columns="n">
内容
</table>
```

下面是元素table各个属性的功能和用法介绍。

(1) align。该属性用于指定表各个列中文本和图像的对齐方式,其值为c、l和r,分别表示居中、左对齐和右对齐。

(2) title。该属性用于指定table元素的标题。

(3) columns。该属性用于指定表格的列数,该数不能为0。

值得注意的是:元素td必须位于元素tr内,而元素tr必须位于table元素内。下面是表格的范例。

【例13-4-3】 (代码位置:\13\testTable.wml)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card title="Table">
<p>
  <table columns="3" border="1" cellspacing="0" >
    <tr>
      <td>(行 1,单元格 1)</td>
      <td>(行 1,单元格 2)</td>
      <td>(行 1,单元格 3)</td>
    </tr>
    <tr>
      <td>(行 2,单元格 1)</td>
      <td>(行 2,单元格 2)</td>
      <td>(行 2,单元格 3)</td>
    </tr>
  </table>
</p>
</card>
</wml>
```

范例运行结果如图 13-4-1 所示。

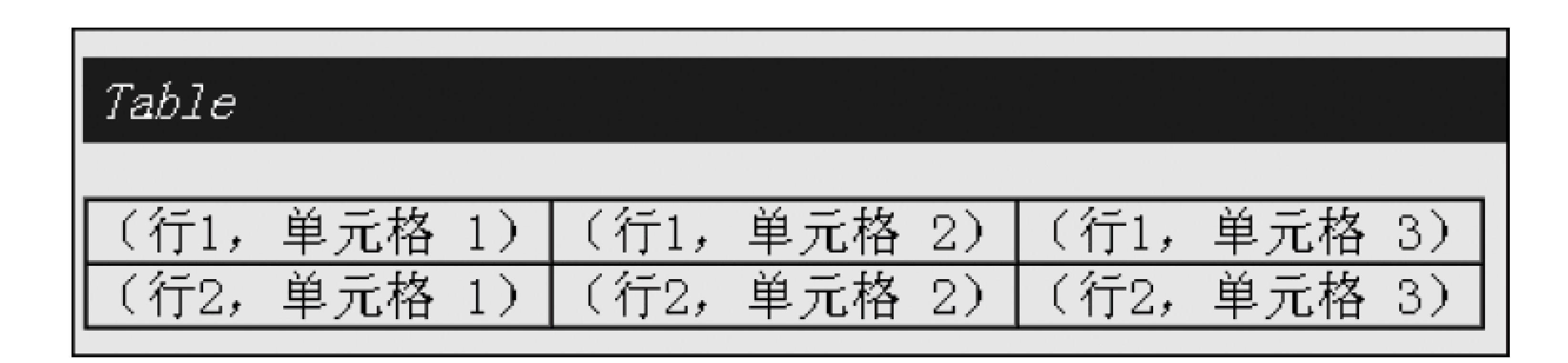


图 13-4-1　表格

13.4.7　文本格式化元素

WML 定义了很多供格式化输出的元素,比如粗体字。表 13-4-1 给出可以影响文本格式的元素。

表 13-4-1　文本格式化元素

元素名	<b>	<big>	<em>	<i>	<small>	<strong>	<u>
元素描述	定义粗体的文本	定义大号的文本	定义着重文本	定义斜体的文本	定义小号的文本	定义强调文本	定义下划线文本

下面是文本格式化元素的范例。

【例 13-4-4】　(代码位置:\13\testFormat.wml)

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card title="文本格式化范例">
<p>
  <em>着重文字</em><br/>
  <strong>强调文字</strong><br/>
  <b>加粗文字</b><br/>
  <i>斜体字文字</i><br/>
  <u>下划线文字</u><br/>
  <big>大号文字</big><br/>
  <small>小号文字</small>
</p>
</card>
</wml>
```

图 13-4-2 文本格式化

范例运行结果如图 13-4-2 所示。

13.4.8 image 元素

WAP 协议允许使用一种叫做 WBMP 的图形标准，它是一个位（黑或者白）的 BMP 版本。因此，所有的图形为了适合这个协议必须转换成 WBMP。在使用 WBMP 的时候当然有很多限制，图像的大小不能超过 150×150 像素。这是因为 WAP 设备的屏幕都非常有限。另外受 WAP 电话的内存限制，图像的大小不能超过 1461B。关于图像转换成 WBMP 格式图像，RCP 分发系统有 Adobe Photoshop 5.x 和 PaintShop Pro 的插件支持 WBMP 格式。WML 提供标签<img>用来定义图像，图像的格式必须是 WBMP，标签格式如下：

```
<img src="somefile.wbmp" alt="sometext" width="width" height="height"/>
```

属性 width 和 height 表示图像宽度和高度，单位为像素；alt 表示图像无法显示时使用的代替文字；scr 表示 wbmp 的 URML。scr 是必选项，其他的是可选项。

13.4.9 锚和任务元素

链接是 HTML 页面里最基本的功能，在 WML 里也一样用和来建立链接的内容，必选属性 href 指定了要打开的 URL，可选的 title 属性给该链接取个标记名字。

然而链接只是 WML 中的一种任务，WML 定义有很多不同类型的任务。WML 允许用户单击（或其他事件发生）时做各种任务，这些任务可以是链接、刷新页面、打开历史 URL 和空操作等任务。WML 使用<go>、<prev>、< refresh>和<noop>等标签来定义这些任务。为了能够让用户或事件激活使用这些任务，WML 提供了<anchor>和<do>等标签用来绑定任务。<anchor>标签将一个

格式化的文本和一个任务捆绑在一起，经常被称作超级链接。标签<do>将用户的交互(例如功能键、声音提示等)和一个任务联系在一起，当用户激活这些功能的时候，就执行和<do>相联系的任务。下面是用锚标签<anchor>定义链接任务的范例，范例在<anchor>中定义有<go>和<a>标签，这两个标签都表示链接任务，前者任务链接到 test1. wml，而后者任务链接到 test2. wml。

【例 13-4-5】 (代码位置：\13\testAnchor. wml)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/
wml_1.1.xml">
<wml>
<card>
<p>
<anchor title="Link1"><go href="test1.wml"/>News</anchor><br/>
<a title="Link2" href="test2.wml">Sports</a>
</p>
</card>
</wml>
```

范例并没有给出文件 test1. wml 和 test2. wml，因此单击 News 和 Link2 链接时浏览器将报告链接文件找不到的提示。下面介绍 WML 任务元素。

1. 元素 go

<go>是一个任务标签，用来指示浏览器显示指定的 URL，包括 deck 的第一个 card 或者指定的 card。语法如下：

```
<go href="url" sendreferer="false|true" method="get|post" accept-charset=
"charset">
<postfield name="name" value="value"/>
<setvar name="name" value="value"/>
</go>
```

其中属性 href 为必选，其他为可选，href 用于指定目标 URL 地址。属性 sendreferer 表示是否传递调用 href 所指定的 URL 的页面的 URL，也就是当前页的 URL，即 HTTP 头中的 HTTP_REFERER，默认值 false 表示不传递，可选值为 true。属性 method 和 HTML 中表单的 method 属性一样，指定表单是以 get 的方式还是 post 的方式递交，以便 CGI 处理，默认值为 get，但如果未指定 method 而<go></go>间存在<postfield>，手机会自动以 post 方式传递。属性 accept-charset 可覆盖在 HTTP 头中指定的字符集，可以写多个字符集，如 accept-charset＝"UTF-8，US-ASCII，ISO-8859-1"。

<postfield name＝"name" value＝"value"/>可以看作是 HTML 表单中的<INPUT TYPE＝"HIDDEN" NAME＝"变量名"VALUE＝"值">，通过它可以向指定

的URL传递“变量名/值”形式的数据。name和value属性都是必选的。注意，只有这里的变量是用来递交给CGI程序的。

除了<postfield>，还可以在<go>和</go>间加入一句或者多句<setvar name="name" value="value"/>，该语句的意思是，当触发某一事件时给变量赋值。

要注意的是，当<go>和</go>之间没有任何语句时，要用<go/>的形式，这点比较特别，比如<anchor title="Link1"><go href="test.wml"/>News</anchor>。元素go的使用可参看上面的例子。

2. 元素 prev

<prev>是一个任务元素，用来将当前页面的URL压入URL历史堆栈，并打开此前的URL，若该URL不存在，则<prev>无效。其格式如下：

```
<prev><setvar name="name" value="value"/></prev>
```

<prev>和</prev>之间可加入一句或多句<setvar name="name" value="value"/>，若不加，则必须变成<prev/>的形式。当用<go>、<prev>和<refresh>标签执行任务时可用标签<setvar>指定变量的值。

下面的例子使用<do>标签向卡片添加一个Back链接，当单击Back按钮会链接到之前的卡片并刷新变量x使其值为30。

【例13-4-6】（代码位置：\13\testDo.wml）

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p>
  <do type="accept" label="Back">
    <prev>
      <setvar name="x" value="30"/>
    </prev>
  </do>
</p>
</card>
</wml>
```

3. 元素 refresh

<refresh>是一个任务元素，用来刷新当前的页面，从而使得页面内的变量刷新，其语法如下：

```
<refresh><setvar name="name" value="value"/></refresh>
```

下面是刷新页面的任务范例范，范例由Refresh.wml和displayVar.wml文件组成，

Refresh. wml 负责刷新设置页面变量 x 的值，displayVar. wml 负责显示变量 x。

【例 13-4-7】 （代码位置：\13\RefreshTask\目录）

文件 Refresh. wml 内容如下：

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.
1.xml">
<wml>
<card>
<p>
<anchor>刷新页面
<go href="displayVar.wml"/>
<refresh><setvar name="x" value="30"/>
</refresh>
</anchor>
</p>
</card>
</wml>
```

文件 displayVar. wml 内容如下：

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.
1.xml">
<wml>
<card id="card0" ordered="false" title="变量 X">
<p>
$(x)
</p>
</card>
</wml>
```

4. 元素 noop

<noop>是一个任务元素，表示什么也不做，该标签不能用在<anchor>中，一般用在覆盖 deck 级的<do>中。下面的例子使用<do>标签向卡片添加一个 Back 链接，但是<noop>标签会阻止此操作，当单击 Back 链接时什么也不会发生。

【例 13-4-8】 （代码位置：\13\testNoop. wml）

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p>
```

```
    <do type="prev" label="Back">
      <noop/>
    </do>
  </p>
  </card>
  </wml>
```

13.4.10 input 元素

元素 input 用于定义输入字段，它对输入文本内容的格式、数据类型、长度、值和变量名等多种属性有具体的规定。input 元素是 WML 编程中处理用户交互活动的重要元素，它通过单独的<input>标签进行定义，其语法格式如下：

```
<input name="variable" title="label" type="type" value="value" default="default"
format="specifier" emptyok="false|true" size="n" maxlength="n" tabindex="n"/>
```

其中除了 name 属性是必选的以外，其他属性都是可选的。这些属性的功能和用法如下所示。

（1）name。该属性指定了用来存储用户输入文本的变量名称。

（2）title。该属性指定输入框的标题。

（3）type。用于指定文本输入区的类型，有 text 和 password 两种选择。默认值为 text，指定的用户可以输入文本，而且输入的文本会同时逐渐响应并显示在浏览器中。如果选择 password，则指定用户输入的文本作为密码文本处理，WML 程序按文本实体接收输入的数据，而浏览器上响应用户输入显示时均为星号“*”，由此起到保密的目的。

（4）value。该属性用于存储用户输入的文本值，该值可通过属性 name 指定的变量名访问，该值将显示在输入框中。

（5）default。该属性用于指定属性 name 所定义变量的默认值。

（6）format。该属性用于格式化输入的数据。默认是"*M"。值 A 表示大写字母或标点符号，值 a 表示小写字母或标点符号；值 N 表示数字；值 X 表示大写字符，值 x 表示小写字符；值 M 表示所有字符，值 m 表示所有字符；值 *f 表示任何数量的字符。用上面的字母替代 f，可以规定用户可输入的字符类型；值 nf 表示用数字 1～9 代替 n，可以规定用户可输入的字符数目。用上面的字母替代 f，可以规定用户可输入的字符类型。

（7）maxlength。该属性用于指定用户可输入字符串的最大长度。该属性的上限为 256，最多不能超过 256 个字符。

（8）emptyok。用于指定用户是否可以在输入框内输入内容。

（9）size。该属性用于指定输入框的宽度，宽度值为字符个数。

（10）tabindex。用于指定多个输入框存在时的焦点次序。

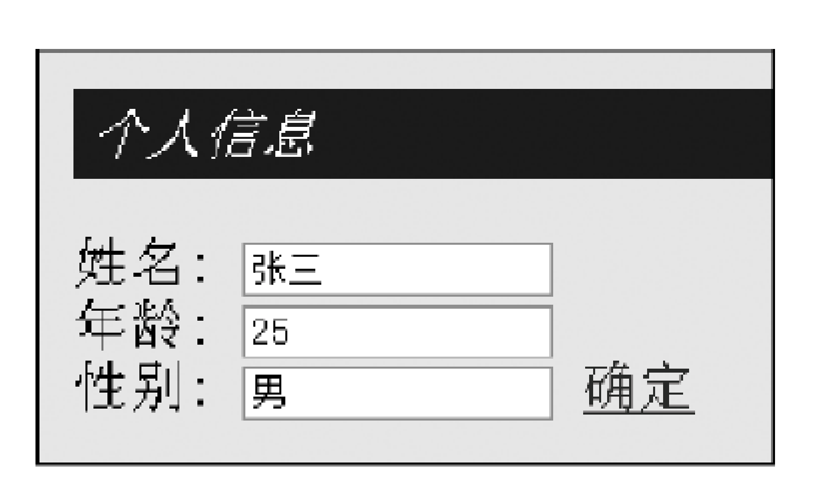
下面是元素 input 的范例，card1 卡片会先使用 input 元素接收用户输入的姓名、年龄和性别，用户单击“确定”按钮后会跳转到 card2 卡片，card2 卡片会显示出 card1 卡片中的姓名、年龄和性别。

【例 13-4-9】（代码位置：\13\testInput.wml）

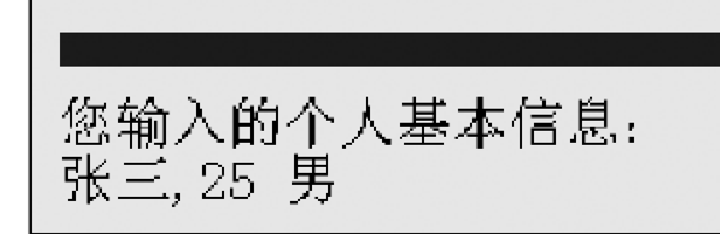
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card title="个人信息" id="card1">
<p>
  姓名：<input name="name" size="15"/><br/>
  年龄：<input name="age" size="15" format="*N"/><br/>
  性别：<input name="sex" size="15" format="M" maxlength="1"/>
  <a href="#card2">确定</a>
</p>
</card>
<card id="card2">
您输入的个人基本信息：<br/>
$(name),$(age)$(sex)
</card>
</wml>
```

程序中$(name)、$(age)和$(sex)表示变量，分别对应card1中的三个文本框，“年龄”字段限输入数字，“性别”字段只限输入一个字符。运行结果如图13-4-3所示，其中图13-4-3(a)是个人信息输入界面，在相应字段中输入了“张三”、“25”和“男”，单击“确定”按钮后浏览器输出图13-4-3(b)的内容。如果“年龄”字段中输入的不是数字，微型浏览器程序忽略无效输入格式。图13-4-4是非法输入年龄字段的运行结果。



(a) 个人信息输入界面



(b) 个人信息显示

图 13-4-3 合法输入的运行结果

(a) 个人信息输入界面

(b) 个人信息显示

图 13-4-4 非法输入的运行结果

13.4.11 select 元素

选择列表属于输入元素，允许用户从选项列表中选择需要的项目。WML 不仅支持单选列表，即单选项，而且支持多选列表，即复选项。select 元素允许用户从选项列表中选择所需的项目。列表中的选项由 option 元素进行定义，一般是一行格式化的文本。编程时，可以使用 optgroup 元素将 option 元素的情况项目分成不同级别或层次的选项组，为用户选择提供方便。select 元素是通过<select>和</select>标签进行定义的，语法格式如下：

```
<select title="label" multiple="false|true" name="variable" default=
"default" iname="index_var" ivalue="default" tabindex="n">
内容(content)
</select>
```

其中所有属性都是可选的。select 元素各个属性的功能和用法如下。

（1）multiple。用于指定选择列表是否可以使用复选框。

（2）name。用于指定接收选项值的变量的名称，变量值由 value 属性预设定。

（3）value。用于指定 name 属性所定义变量的默认值。

（4）iname。用于指定包含排序号的变量的名称。

（5）value。用于指定选择列表中被选中选项的值，是一个具有排序号性质的值。

（6）title。用于指定选择列表的标题。

（7）tabindex。用于指定当前选择光标在选择列表中的具体位置，该位置即当前选择操作将要选择的选项所在的位置。

下面是选择元素的范例，例中有两个选择列表，一个是单选列表，一个是多选列表。

【例 13-4-10】（代码位置：\13\testSelect.wml）

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.
1.xml">
<wml>
<card id="card0" ordered="false" title="选择">
<p>
请选择城市：
<select title="城市列表" name="city">
<option title="北京" value="c1">北京</option>
<option title="上海" value="c2">上海</option>
<option title="香港" value="c3" onpick="http://127.0.0.1/helloworld.wml">香港</option
>
</select>
请选择列：
<select title="Column List" multiple="true" name="col">
<option title="热点新闻" value="l1">新闻</option>
```

```
<option title="酷运行" value="12">运行</option>
<option title="Pop Enter,,," value="13">流行元素</option>
</select>
</p>
</card>
</wml>
```

范例运行结果如图 13-4-5 所示，图左边的是单选列表，右边的是多选列表。当用户在单选列表中选中“香港”选项时，系统会跳转至 http://127.0.0.1/helloworld.wml。

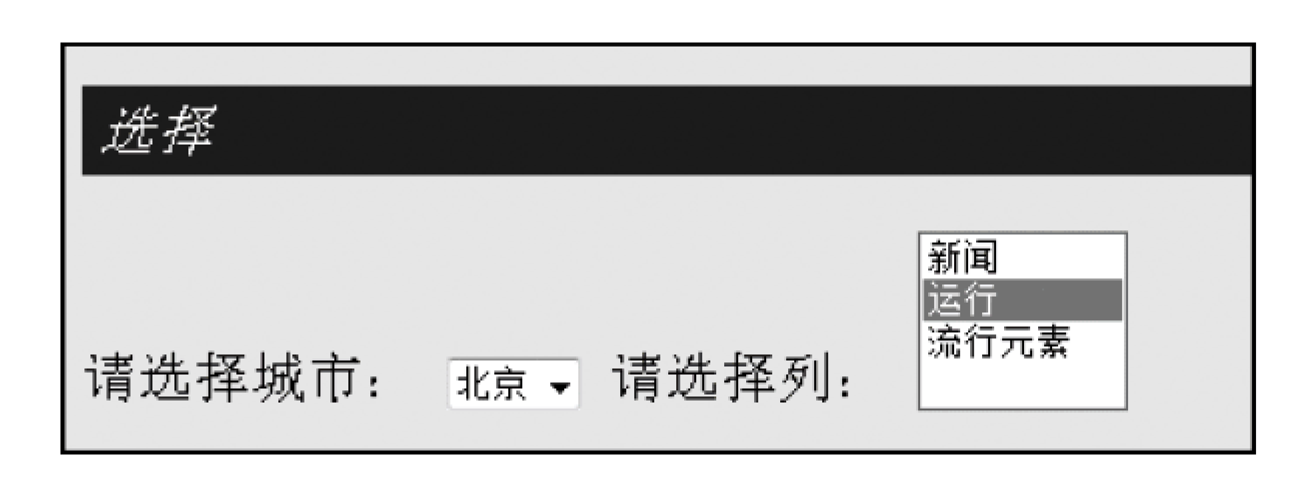


图 13-4-5 选择元素

13.4.12 option 元素

option 元素用于定义 select 元素中的一组单选项。它通过<option>和</option>标签进行定义，并可包括事件和单选项的显示文本等信息，其语法格式如下：

```
<option title="label" value="value" onpick="href">
内容
</option>
```

option 元素的属性均为可选，各属性功能及用法说明如下。

（1）value。该属性用于设置键值。当用户选到该选项之后，option 元素就会将该值赋给 select 元素的 name 属性所指定的变量。

（2）title。用于 option 元素指定的一个标题，以便提示用户操作。

（3）onpick。当用户选择项目时发生的事情，其值是 URL。

13.5 PHP 在 WAP 开发中的应用

可以使用 PHP 来编写 WAP 程序，比如开发动态的 WML 页面，不过条件是需要将输出的标签或语句限制在 WAP 微型浏览器可接受的范围之内。而且，PHP 还可以在一个 HTML(HDML)文件中编写出既适合于 HTML(HDML)，也适合于 WML 的内容。PHP 的源代码对于 WAP 客户端来说是不可见的，所以可以针对 HTML 浏览器输出 HTML 页面，而针对 WAP 浏览器输出 HDML 页面或 WML 页面。

13.5.1 基本规则

使用 PHP 编写 WML 程序应该要遵循基本的规划，具体基本规则如下。

(1) 使用“<?”和“?>”来包含 PHP 编写的 WML 程序行，格式如下：

```
<?
    程序行
?>
```

(2) 程序行语句均以分号(；)结尾，这是与 ASP 编写 WML 程序不同的地方。

(3) 使用 PHP 的 header 关键字来声明 WML 的文件类型，基本格式为：

```
header("Content-type: text/vnd.wap.wml");
```

上述格式是专为 WAP 浏览器识别而采取的书写格式。如果开发中用户想使用普通的浏览器，如 IE 浏览器来测试程序效果，则可在该语句的前面加上双斜线(//)，格式如下：

```
//header("Content-type: text/vnd.wap.wml");
```

这样，基于 PC 的浏览器将忽略程序中无法理解的 WML 标签，这时当前的程序实际上成为了 HTML 页面。当想在 WAP 设备或者模拟器上测试的时候，只需要去掉//，当前页面就会自动变成 WML 页面。

(4) 其他 WML 标签和语句行一律使用 PHP 的 echo 关键字来声明，每个 WML 行的后面加上换行符\n(根据需要，也可加多个)。其基本格式如下：

```
echo("…… (WML 标签和语句行)\n");
```

例如，下面的语句就是 PHP 编写 WML 语句行的典型例子：

```
echo("<?xml version=\"1.0\"?>\n");
```

而且，echo 语句中的“WML 标签和语句行”可以连写，以节省程序代码行。例如，下面的一行语句就连写了 WML 的 3 行标签语句：

```
echo("<wml><card><p>");
```

(5) 声明脚本语言。这是对 PHP 编程的继承，向编译器声明当前程序采用的脚本语言为 PHP，格式如下：

```
<script language="PHP">
```

不过这一句可以省略，因为编译器能够自动识别 PHP 的脚本程序。

(6) 文件最后保存时采用的扩展名为.php，而不是.wml 等扩展名。

根据上述规则，可以用 PHP 编写出 WML 页面的文件头，并让程序在 WAP 浏览器中显示“Hello，World!”。下面是 PHP 在 WAP 应用中的范例。

【例 13-5-1】 (代码位置：\13\testPHPW_Hello.wml)

```
<?
    header("Content-type: text/vnd.wap.wml");
    echo("<?xml version=\"1.0\"?>\n");
```

```
    echo("<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\" \"http://www. wapforum.
    org/DTD/wml_1.1.xml\">\n\n");
?>
<?
    echo("<wml><card><p>\n");
    echo("Hello,World!\n");
    echo("</p></card></wml>\n");
?>
```

13.5.2 PHP实现WAP程序

下面给出一个非常有用的由PHP实现的WML应用范例，它被称为PizzaCalc，可以根据输入的pizza账单及人数算出每个人的花费。注意，程序中根据WML的规则，对一些特殊符号（如双引号等）使用了转义字符。该程序不仅可以进行变量处理，而且还可以传递相应的参数。范例程序代码如下。

【例13-5-2】 （代码位置：\13\testPHPW_pizzacalc. php）

```
<?
  header("Content-type: text/vnd.wap.wml");
  echo("<?xml version=\"1.0\"?>\n");
  echo("<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\" \"http://www.wapforum.org/
  DTD/wml_1.1.xml\">\n\n");
?>

<?
  echo("<wml>\n");
?>

<?
  if($action=="calc"){
    echo("<card id=\"result\" title=\"PizzaCalc\">\n");
    echo("<do type=\"prev\" label=\"Back\">\n");
    echo("<go href=\"pizzacalc.html#input\"/>\n");
    echo("</do>\n");
    echo("<p>\n");
    echo("The cost per eater will be ".$total / $eaters."<br/>\n");
  }
  else {
    echo("<card id=\"input\" title=\"PizzaCalc\">\n");
    echo("<p>\n");
    echo("<anchor>Split Pizza bill <go href=\"pizzacalc.html?total=\$(total)&eaters=\
    $(eaters)&action=calc\"/></anchor>\n");
    echo("<br/>\n");
```

```
        echo("Total cost: <input type=\"text\" name=\"total\" format=\"*N\"/>\n");
        echo("Eaters: <input type=\"text\" name=\"eaters\" format=\"*N\"/>\n");
    }
?>

<?
    echo("</p></card></wml>\n");
?>
```

第 14 章 chapter 14

PHP 在 WAP 中的应用

14.1 PHP 动态生成 WAP 页面以及与 WAP 混合开发

PHP 与 MySQL 是目前最流行的网页编程语言和数据库的搭配。它们属于 WWW 或者有线互联网的范畴，由于使用方便与功能强大的特点而美名传播，那么它们能够为 WAP 服务吗？答案是肯定的。不熟悉 WAP 或者 PHP 的人或许有些不解，毕竟在传统意义上 PHP 提供的是 WWW 的内容与服务，对于 WAP，它们能“兼容”吗？事实上 PHP 是能为 WAP 服务的。一个典型的 WAP 应用请求步骤描述如下。

（1）具有 WAP 用户代理功能的移动终端（如 WAP 手机），通过内部运行的微浏览器向某一网站发送 WAP 服务请求。该请求先由 WAP 网关截获，对信息内容进行编码压缩，以减少网络数据流量，同时根据需要将 WAP 协议转换成 HTTP 协议。

（2）协议将处理后的请求转送到相应 WAP 服务器。在 WAP 服务器端，根据页面扩展名等属性，被请求的页面直接或由服务器端脚本解释后输出，再经过网关传回给用户。

从上述的 WAP 应用流程可以发现，生成动态 WAP 页面与动态产生 Web 网页的过程非常相似。但是由于 WAP 应用使用的 WML 语言来源于语法严格的 XML，因此要求输出的格式必须按 WAP 网页的规范输出。同时，由于 WAP 协议的应用范围及移动客户端的软、硬件配置等局限性，对每次输出的页面的大小、图像的格式及容量都有一定限制。本章以 PHP 语言为例给出 PHP 动态输出 WAP 页面的实例，以及给出 PHP 与 WAP 混合开发的实例。不过要注意，由于需要 PHP 解释器来解释该程序并输出 WAP 页面，因此所有类似的程序应以.php 为扩展名。下面是 PHP 动态输出 WAP 页面的实例。

【例 14-1-1】（代码位置：\14\testPHPTOWap.php）

```
<?PHP
header("Content-type: text/vnd.wap.wml");            //定义输出文档为 WML 类型
echo "<?xml version=\"1.0\"?>";
echo "<wml>";
echo "<card title=\"HELLO WORLD\">";
echo "<p>Hello World</p>";
echo "</card>";
```

```
echo "</wml>";
?>
```

下面是PHP和WAP混合编程的实例。

【例14-1-2】 （代码位置：\14\testPHPWAP.php）

```
<?PHP
  header("Content-type: text/vnd.wap.wml");        //定义输出文档为WML类型
  echo "<?xml version=\"1.0\"?>";
?>
<wml>
<card title="Hello World">
<?PHP
echo "<p>Hello World</p>";
?>
</card>
</wml>
```

上面两个写法等价，程序可以在WAP手机模拟器中浏览，输出一句经典的“Hello World”语句，但是在普通的网络浏览器中是无法识别的，原因很简单，在程序开头声明了该输出文档为WML类型，只有WAP设备能够识别并解释。不过还要提醒一句：常见的HTML语言对规范性要求不严，大多数浏览器能“宽容”地接受其中的编写错误；但是WML的规范相当严格，任何错误都可能导致无法输出所需的页面。其运行结果如图14-1-1所示。



图14-1-1 PHP和WAP混编运行结果

14.2 PHP与WAP表单处理技术

和HTML类似，用户可以通过WAP表单把用户填写或选择的数据提交给服务器处理。WML虽然没有提供提交按钮，但可以使用超链接任务来代替提交按钮功能，在go标签中指定提交的方法，通过postfield标签刷新变量的值。下面是表单处理的范例，由login.wml和login.php两个文件组成。文件login.wml用来设计WAP登录界面，它通过表单把用户数据提交给login.php处理。login.php通过变量$_REQUEST获取用户数据，判断用户是否合法并给出相应的信息。

【例14-2-1】 （代码位置：\14\PHPWAPForm\目录）

文件login.wml内容如下：

```
<?xml version="1.0"?>
<wml>
<card title="登录模块(用户名:admin,密码:11111111)">
<p>
```

```
    用户名：<input name="username" type="text"/><br/>
    密　码：<input name="pwd" type="password"/>
    <do type="text" label="确定">
    <go method="get" href="login.php">
    <postfield name="username" value="$(username)"></postfield>
    <postfield name="pwd" value="$(pwd)"></postfield>
    </go>
    </do>
</p>
</card>
</wml>
```

文件 login.php 内容如下：

```
<?PHP
    header("Content-type: text/vnd.wap.wml");          //定义输出文档为 WML 类型
    echo "<?xml version=\"1.0\"?>";
?>
<wml>
<card title="Hello World">
<?PHP
$username=$_REQUEST["username"];
$pwd=$_REQUEST["pwd"];

if($pwd=="11111111" && $username=="admin")
      echo "欢迎您,$username";
else echo "用户名或密码出错!";
?>
</card>
</wml>
```

先在微浏览器中输入 http://127.0.0.1/login.wml，然后输入用户名和密码，正确的用户名和密码是 admin 和 11111111，再单击“确定”按钮。程序运行结果如图 14-2-1 所示，其中图 14-2-1(a)是登录界面，图 14-2-1(b)是当用户登录失败时的界面，图 14-2-1(c)是当用户登录成功时的界面。



(a) 登录界面

Hello World
用户名或密码出错!

(b) 登录失败

Hello World
欢迎您，admin

(c) 登录成功

图 14-2-1　WAP 登录

14.3　PHP 与 WAP 图像处理技术

WAP 使用的图像是一种特殊的黑白图像格式：WBMP。开发者可以使用一些现有的工具将一般图像转换成 WBMP 格式，然后在 WML 文档中使用。但是如果在 WAP 程序中能动态地生成所需图像（如股市的 K 线图），将会使程序有极其广阔的应用前景。PHP 提供了强大的图形绘制功能，有关 PHP 图像的处理可参看前面的有关章节或 GD2 图像库相关资料。下面是 PHP 处理 WAP 图像的范例，实例会在 WAP 模拟器中显示两个黑色的矩形框，例中由 rectangle. wml 和 rectangle. php 两个文件组成，前者显示图像，后者生成 WBMP 格式图像。

【例 14-3-1】（代码位置：\14\testPHPWAPPic 目录）

文件 rectangle. wml 内容如下：

```
<?xml version="1.0"?>
<wml>
    <card id="start" title="rectangle">
        <p align="center">
            <img src="rectangle.php" alt="rectangle"/>
        </p>
    </card>
</wml>
```

文件 rectangle. php 内容如下：

```
<?
header("content-type: image/vnd.wap.wbmp");
$im=imagecreate(90,90);
$white=imagecolorallocate($im,255,255,255);
$black=imagecolorallocate($im,0,0,0);
imagerectangle($im,5,5,85,85,$black);
imagerectangle($im,15,15,75,75,$black);
imagewbmp($im);
imagedestroy($im);
?>
```

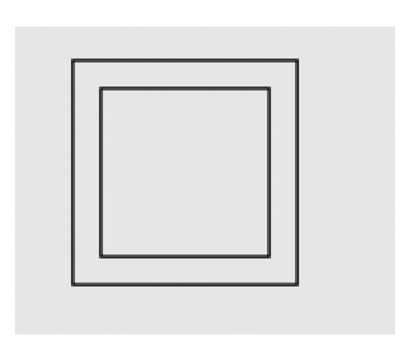


图 14-3-1　输出两个方框

在微浏览器中输入 http://127. 0. 0. 1:82/rectangle. wml，程序运行后其结果如图 14-3-1 所示。

下面是验证码图像生成的范例，由 checkCode. wml 和 checkCode. php 两个文件组成，前者用于显示图像，后成在后台生成 WBMP 格式验证码图像，图像中的验证码长度为 4，其内容和位置是随机的，图像同时还伴有若干黑色的干扰像素。

【例 14-3-2】（代码位置：\14\testWAPCheckCode\ 目录）

文件 checkCode. wml 内容如下：

```
<?xml version="1.0"?>
<wml>
    <card id="start" title="CheckCode">
        <p align="center">
            验证码：<input name="checkcode"/><img src="checkCode.php" alt=
            "rectangle"/>
        </p>
    </card>
</wml>
```

文件 checkCode.php 内容如下：

```
<?php
  class Vailimg {
      private $width;                              //验证码图片的宽度
      private $height;                             //验证码图片的高度
      private $codeNum;                            //验证码字符的个数
      private $checkCode;                          //验证码字符
      private $image;                              //验证码画布

      //构造方法用来实例化验证码对象,并为一些成员属性初始化
      //参数 width: 设置验证码图片的宽度,默认宽度值为 60 像素
      //参数 height: 设置验证码图片的高度,默认高度值为 20 像素
      //参数 codeNum: 设置验证码中字母和数字的个数,默认个数为 4 个
      function __construct($width=60,$height=20,$codeNum=4){
          $this->width=$width;                     //为成员属性 width 初使化
          $this->height=$height;                   //为成员属性 height 初使化
          $this->codeNum=$codeNum;                 //为成员属性 codeNum 初使化
          $this->checkCode=$this->createCheckCode();         //为成员属性 checkCode 初使化
      }
      function showImage(){
          //通过访问该方法向浏览器中输出图像
          $this->getCreateImage();                 //生成验证码图像
          $this->outputImage();                    //生成相应格式的图像并输出
      }
      function getCheckCode(){
          //访问该方法获取随机创建的验证码字符串
          return $this->checkCode;               //返回成员属性$checkCode 保存的字符串
      }
      private function getCreateImage(){
          //用来创建图像资源,并初使化背影
          $this->image=imagecreate($this->width,$this->height);
          $white=imagecolorallocate($this->image,255,255,255);
          $black=imagecolorallocate($this->image,0,0,0);
          imagerectangle($this->image,1,1,$this->width-1,$this->height-1,$black);
```

```
        //随机颜色、随机摆放的随机字符串向图像中输出
        for($i=0;$i<$this->codeNum;$i++){
            $x=floor($this->width/$this->codeNum)*$i+3;
            $y=rand(0,$this->height-15);
            $c=$this->checkCode[$i];
            imagechar($this->image,5,$x,$y,"$c",$black);
        }

        //设置干扰像素,向图像中输出不同颜色的 50 个点
        for($i=0;$i<50;$i++){
            $color=rand(0,1);
            switch($color)
            {
                case 0:
                    imagesetpixel($this->image,rand(1,$this->width-2),rand(1,$this
                    ->height-2),$black);
                    break;
                case 1:
                    imagesetpixel($this->image,rand(1,$this->width-2),rand(1,$this
                    ->height-2),$white);
                    break;
                default: break;
            }
        }
    }
    private function createCheckCode(){
    //随机生成用户指定个数的字符串
        for($i=0;$i<$this->codeNum;$i++){
            $number=rand(0,2);
            $rand_number=48;
            switch($number){
                case 0 : $rand_number=rand(48,57);break;        //数字
                case 1 : $rand_number=rand(65,90);break;        //大写字母
                case 2 : $rand_number=rand(97,122);break;       //小写字母
                default: break;
            }
            $ascii=sprintf("%c",$rand_number);
            $ascii_number[$i]=$ascii;
        }
        return $ascii_number;
    }

    private function outputImage(){
    //自动检测 GD 支持的图像类型,并输出图像
```

```
        //判断生成 WBMP 格式图像的函数是否存在
        header("content-type: image/vnd.wap.wbmp");   //发送标头为 image/wbmp
        imagewbmp($this->image);                         //以 WBMP 格式将图像输出到浏览器
        }
        function __destruct(){
        //在对象结束之前销毁图像资源,释放内存
        imagedestroy($this->image);
        //调用 GD 库中的方法销毁图像资源
        }
}

$img=new Vailimg(70,20,4);                    //初始化画布宽高为 70*40,验证码长度为 4
$img->showImage();                            //显示验证码图像
?>
```

在微浏览器中输入 http://127.0.0.1/checkCode.wml,程序运行后其结果如图 14-3-2 所示。

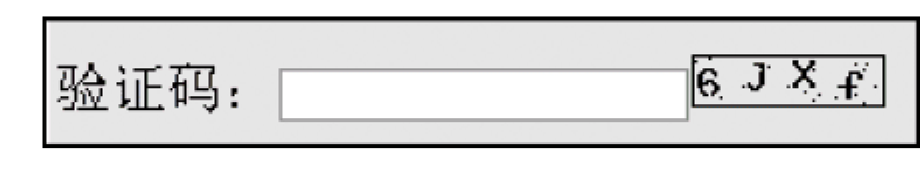


图 14-3-2　验证码图像

上面两个实例都用到了函数 imagewbmp,该函数的作用是以 WBMP 格式将图像输出到浏览器或文件。其格式如下:

```
bool imagewbmp(resource $image [,string $filename [,int $foreground]])
```

函数用 image 图像创建一个名为 filename 的 WBMP 文件。参数 image 是函数 imagecreatetruecolor 的返回值。filename 参数是可选项,如果省略,则直接将原图像流输出。通过用函数 header 发送 image/vnd.wap.wbmp 的 Content-type,可以创建直接输出 WBMP 图像的 PHP 脚本。

14.4　PHP 与 WAP 汉字处理技术

作为一种全球应用的协议,WAP 选择了 Unicode 2.0 作为其标准字符集编码,能够同时处理英、中、日、法等多种文字。但是开发者日常处理的汉字是 GB 2312 编码,不同的内码标准必定不能通用,因此,如果不在两种编码之间进行码制的转换,就会出现汉字乱码的现象。目前的大多数 WAP 手机(Nokia7110、爱立信 R320S 等)都使用 UTF-8(即 Unicode)来编码。如果直接在 WML 使用中文字符(GB 2312 编码),将会产生乱码,导致手机用户无法识别,因此在输出中文之前,必须使用程序或函数(关于此类的 PHP 函数库,网络上已有非常多技术成熟的产品可以下载)对中文进行 Unicode 的编码。而在少数支持 GB 2312 编码的手机或 WAP 终端设备中,开发者只需在程序中定义文档的内码类型后,即可直接正确显示汉字。下面是 PHP 处理 WAP 汉字的范例。

【例 14-4-1】 (代码位置:\14\testWAPGB312.php)

```
<?PHP
```

```
    header("Content-type: text/vnd.wap.wml;charset=gb2312");    //定义输出文档为 WML 类型
    echo "<?xml version=\"1.0\"?>";
    echo "<wml>";
    echo "<card title=\"中文测试\">";
    echo "<p>你好！</p>";
    echo "</card>";
    echo "</wml>";
?>
```

在程序的 header 语句中，定义了文字的编码为 GB 2312，如果用户的手机支持 GB 2312 编码，将会显示“你好”的字样。但值得注意的是，除了在 header 语句中定义文字的编码方 GB 2312，同时，文件本身要用 UTF-8 或 GB 2312 编码保存。

14.5 WAP 中的 Session 技术

WAP 与 PC 不同之处在于 WAP 1.x 版本不支持 Session 与 Cookie，WAP 2.0 虽然支持这些，但是只有协议支持还是不行，还要看手机终端是否支持。在手机网站使用 Session 时，在最老型号的手机和 m3gate opera9 上都不支持接收 Session，所以就会造成 m3gate 显示第一行第一列有致命错误。现在很多 WAP 社区和 3G 门户等都没有解决这一问题。关于 Session 可以有两种解决方法，一种是完全不依赖 WAP 协议，完全不依赖手机终端，采用对 URL 进行重写的办法来支持 Session，这个方法只与服务器端有关。另一种是在输出 session_start 前使用 session_cache_limiter("none")，这样就不会下载任何东西到本地，而又能支持 Session。

下面是在 WAP 网站中 PHP 通过函数 session_cache_limiter 使用 Session 的范例，例中通过 Session 记住用户变量 $timer，从而能实现变量的递增，有关 Session 的内容可参看前面章节。

【例 14-5-1】 （代码位置：\14\testWAPSession.php）

```
<?PHP
session_start();
session_cache_limiter("none");
header("Content-type: text/vnd.wap.wml");          //定义输出文档为 WML 类型
echo "<?xml version=\"1.0\"?>";
echo "<wml>";
echo "<card title=\"Session\">";

if(isset($_SESSION["timer"]))
    $timer=$_SESSION["timer"];
else $timer=0;
echo "<p>";
echo $timer;
echo "</p>";
```

```
$timer++;
$_SESSION["timer"]=$timer;
echo "</card>";
echo "</wml>";
?>
```

接下来对重写URL实现Session方法做一个介绍。通过session_start启动Session后，首先会检查发出请求页面的Cookie、URL和表单中是否存在一个名称为PHPSESSID的变量，也就是说在$_COOKIE、$_GET和$_POST三个数组中是否有PHPSESSID这个变量。如果存在，就去相应目录中读取以sess_开头、加上这个Cookie变量值为文件名的文本文件，并从这个文本文件中取得已经设置好的数据，添加到$_SESSION数组中去，这样，就可以通过$_SESSION来取得Session变量。可以把PHPSESSID保存在URL中，在输出的每个网页中对超链接部分做一个替换，给每个超链接加上一个PHPSESSID=×××这么一个变量，这样就可以使用Session来传递一些变量了。在用户第一次访问WAP网站的时候，自己生成一个PHPSESSID值，然后把这个值附在每个超链接的后面，下次检查URL中如果有PHPSESSID参数，则不再生成PHPSESSID值。按照上面的方法就可以在WAP系统中任意使用Session值。

14.6 识别客户端请求类型

当服务器收到请求后，需要知道请求类型并依据该类型发送WML或HTML格式的数据给客户端，服务器如何能够知道请求是来自WML浏览器还是HTML浏览器？既然要利用已经存在的为HTML浏览器编写的代码，就需要知道请求是从HTML浏览器还是从WML浏览器过来的。同样地，如果想重新引导HTML浏览器直接到相应的HTML文档上，WML浏览器到WML页面上，下面的PHP代码就可以做到这些。

【例14-6-1】（代码位置：\14\testWAPHTMLRequest.php）

```
<?
//要加载的HTML文件名
    $htmlredirect="my_htmlpage.html";
//WML绝对路径
    $wmlredirect="http://wap.mysite.com/wml/my_wmldeck.wml";
    if(strpos(strtoupper($HTTP_ACCEPT),"VND.WAP.WML")>0)
    {   //检查浏览器或WAP网关是否声明该请求是WML请求
        $br="WML";
    }
    else {
        //判断请求是否来自微浏览器
        $browser=substr(trim($HTTP_USER_AGENT),0,4);
        if(  $browser=="Noki" ||  //Nokia phones and emulators
             $browser=="Eric" ||  //Ericsson WAP phones and emulators
```

```
            $browser=="WapI" ||   //Ericsson WapIDE 2.0
            $browser=="MC21" ||   //Ericsson MC218
            $browser=="AUR " ||   //Ericsson R320
            $browser=="R380" ||   //Ericsson R380
            $browser=="UP.B" ||   //UP.Browser
            $browser=="WinW" ||   //WinWAP browser
            $browser=="UPG1" ||   //UP.SDK 4.0
            $browser=="upsi" ||   //another kind of UP.Browser
            $browser=="QWAP" ||   //unknown QWAPPER browser
            $browser=="Jigs" ||   //unknown JigSaw browser
            $browser=="Java" ||   //unknown Java based browser
            $browser=="Alca" ||   //unknown Alcatel-BE3 browser
            $browser=="MITS" ||   //unknown Mitsubishi browser
            $browser=="MOT-" ||   //unknown browser
            $browser=="My S" ||   //unknown Ericsson devkit browser
            $browser=="WAPJ" ||   //Virtual WAPJAG www.wapjag.de
            $browser=="fetc" ||   //fetchpage.cgi Perl script from www.wapcab.de
            $browser=="ALAV" ||   //yet another unknown UP based browser
            $browser=="Wapa")     //another unknown browser(Web based "Wapalyzer")
            {
                $br="WML";
            }
            else {
                $br="HTML";
            }
    }
    //判断请求是 WML 还是 HTML 请求,如果是 HTML 请求则返回 HTML 格式文件
    //如是 WML 请求则返回 WML 格式文件
    //由于 HTML 是自定义的,所以要求服务器上存在文件 my_htmlpage.html
    if($br=="WML"){
      //强制浏览器加载 WML 文件
      header("302 Moved Temporarily");
      header("Location: ".$wmlredirect);
      exit;
    }
    else {
      //强制浏览器加载 HTML 文件
      header("302 Moved Temporarily");
      header("Location: ".$htmlredirect);
      exit;
    }
?>
```

这个判断是在服务端完成的,PHP 代码将首先查看网关是否接收 text/vnd. wap.

vml MIME 类型。如果不是,将检测前面的字符,查看是否为 WML 浏览器。如果不符合,那么就假设为 HTML 浏览器。如果有新的 WML 浏览器,那么 ID 字符串也要增加。变量 $HTTP_USER_AGENT 保存有用户代理标签,不同的浏览器其标签值不同,可根据该值判断请求是来自手机还是浏览器。如果用户是从浏览器发出请求,该程序将从服务器直接返回文件 my_htmlpage.html,否则从 http://wap.mysite.com/wml/my_wmldeck.wml 返回数据。

14.7 防止从 Cache 中读取 WML 页面

当 WML 页面下载到 WAP 设备后,它将保存在 WAP 设备内存中一段时间,直到时间过期。这样的缓冲内存叫做 Cache。在时间过期之前,设备都是从 Cache 中获取页面,在时间过期之后,页面才从服务器下载,而不是从 WAP 设备的缓存读取。但是有些时候不想让页面从缓存中读取,而是从服务器端读取。一个典型的例子就是当服务器的内容不断在更新的时候,通过在 HTTP 头中加入一定的 Cache 信息,来告诉 WAP 设备该页面将不存储在缓存中。下面的页面在经过 86 400 秒(24 小时)后过期。

【例 14-7-1】 (代码位置:\14\testWAPCache.wml)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
<meta forua="true" http-equiv="Cache-Control" content="max-age=86400"/>
</head>
<card id="expire1day">
<p>This card will live in the cache for a day</p>
</card>
</wml>
```

14.8 防止从 Cache 中读取变量

变量保存在 Cache 中,这样不需要从服务器中读取变量就提高了工作效率。例如,当用户返回到上一个输入卡片,系统从 Cache 中读取卡片中的变量后,用户就不需要重新输入,从而提高了用户体验。但是在某些情况下这会造成一些问题。例如,在 WAP 聊天系统中,如果直接从 Cache 读取保存聊天内容的变量,变量被用了一遍又一遍,那么将永远看不到最新的聊天内容。在 WML 中,<card>标签有一个参数是 newcontext。当 newcontext="true"时清除所有的变量。但是这样也清除了所有导航的历史记录,这意味着 back 按钮不再工作。为了只清除变量,可以告诉浏览器将变量设为空:

```
<setvar name="one_variable" value=""/>
```

```
<setvar name="another_variable" value=""/>
```

但是,不是每个时候都有效果。在某些情况下必须使用一个难以想象的方法来清空变量,即使用 onenterforward 事件,具体如下所示。

```
<onevent type="onenterforward">
    <refresh>
        <setvar name="one_variable" value=""/>
        <setvar name="another_variable" value=""/>
    </refresh>
</onevent>
```

第 15 章 chapter 15

Android 开发概述

15.1 Android 概念和前景

Android 的中文名称是安卓或安致，Android 是 Google 公司于 2007 年 11 月宣布的基于 Linux 平台的开源移动操作系统的名称。Android 不存在任何以往阻碍移动产业创新的专有权障碍，号称是首个为移动终端打造的真正开放和完整的移动软件。Android 操作系统最初由 Andy Rubin 开发，最初主要支持手机。2005 年由 Google 公司收购注资，并组建包括中国移动、摩托罗拉、三星、华为、英特尔和德州仪器等多家无线应用的领军企业组成的开放手机联盟对 Android 进行开发改良，该开放手机联盟成员包正逐渐扩展到平板电脑、电子阅读器、机顶盒及其他领域上。2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。2012 年 7 月数据，Android 占据全球智能手机操作系统市场 59％的份额，中国市场占有率为 76.7％。Android 操作系统已经成为全球最大的智能手机操作系统。

当前的智能手机操作系统存在多个互相竞争的 Android、Symbian、iOS、WebOS 和 Windows Phone 7 等阵营。在 Android 阵营方面，HTC、三星、摩托罗拉、LG、索爱、华硕、宏碁、华为、中兴、夏普、联想、魅族等数百家巨头推出了基于 Android 操作系统的智能手机或平板电脑。在 Symbian 阵营方面，目前仅有诺基亚一家支持，主要有 N 系列、E 系列和 X 系列的手机。在 iOS 阵营方面，仅有苹果公司一家支持，主要有 iPhone 和 iPad。而 WebOS 阵营仅有生产黑莓手机的 RIM 公司一家支持。类似 Android 阵营的还有 Windows Phone 7，但由于 Windows Phone 7 并不是开源系统，很多厂商没有机会参与，仅有几个手机制造商支持，同时没有相对应的平板电脑操作系统，通过 SDK 对比，Windows Phone 7 和 Android 有很大的差距，从目前微软公司的战略和封闭的 API，可以说很难和 Android 抗衡。Android 的快速发展已经使多家其他的手机操作系统被弃用并导致相应的厂商关闭。

Android 是以 Linux 为核心的手机操作平台，作为一款开放式的操作系统，随着 Android 的快速发展，如今已允许开发者使用多种编程语言来开发 Android 应用程序，而不再是以前只能使用 Java 开发 Android 应用程序的单一局面，因而受到众多开发者的欢迎，成为真正意义上的开放式操作系统。

在 Android 中，开发者可以使用 Java 作为编程语言来开发应用程序，也可以通过 NDK

使用 C/C++ 作为编程语言来开发应用程序，也可使用 SL4A 来使用其他各种脚本语言进行编程（如 Python、Lua、Tcl 和 PHP 等），还有其他诸如 Qt（Qt for Android）和 Mono（Mono for Android）等一些著名编程框架也开始支持 Android 编程，甚至通过 MonoDroid，开发者还可以使用 C# 作为编程语言来开发应用程序。另外，谷歌公司还在 2009 年特别发布了针对初学者的 Android Simple 语言，该语言类似 BASIC 语言。而在网页编程语言方面，JavaScript、AJAX、HTML5、jquery、sencha、dojo、mobl 和 PhoneGap 等都已经支持 Android 开发。

15.2 Android 优势

15.2.1 开放性

Android 完全开源，且该平台从底层操作系统到上层的用户界面和应用程序都不存在任何阻碍产业创新的专有权障碍。同时开源的最大好处是，使得 Android 平台会拥有越来越壮大的开发者队伍，并且随着用户与应用的日益丰富，必然会使得 Android 这个崭新的平台走向成熟与稳定。

15.2.2 挣脱运营商的束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商制约，使用什么功能接入什么网络，几乎都受到运营商的控制。由于 Android 的开放性，所以不同的厂商可以根据自身的情况对 Android 平台进行定制与扩展。这样的一个好处便是，对于用户来说，选择一款 Android 的手机将会有更大的抉择空间，更能展现自身的个性化。

15.2.3 丰富的硬件选择

这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出千奇百怪和功能特色各异的多种产品。功能上的差异和特色却不会影响到数据同步甚至软件的兼容。

15.2.4 应用程序间的无界限

Android 打破了应用程序间的界限，开发人员可以将自己开发的程序与本地的联系人、日历、位置信息等很好地联系起来。此外，应用程序可以自申明其功能以供其他应用程序所使用。

15.2.5 无缝结合的 Google 应用

如今叱咤互联网的 Google 已经走过多年历史。从搜索巨人到全面的互联网渗透，地图、邮件和搜索等 Google 服务已经成为连接用户和互联网的重要纽带，而 Android 平

台手机将无缝结合这些优秀的 Google 服务。

15.3 Android 版本

最初的 Android 在正式发行之前会以著名的机器人名称来对其进行命名，它们分别是，阿童木(Android Beta)；发条机器人(Android 1.0)。后来由于涉及版权问题，谷歌改用甜点名为 Android 命名。各种不同版本的 Android 甜点名如下：纸杯蛋糕(Android 1.5)、甜甜圈(Android 1.6)、松饼(Android 2.0/2.1)、冻酸奶(Android 2.2)、姜饼(Android 2.3)、蜂巢(Android 3.0)、冰淇淋三明治(Android 4.0)和果冻豆(Android 4.1)。

15.3.1 Android 1.1

该版本于 2009 年 2 月发布，其主要特性为闹钟、API 示例、浏览器、计算器、摄像头、联系人、开发工具包、拨号应用、电子邮件、地图(包含街景)、信息服务、音乐、图片、设置。

15.3.2 Android 1.5

该版本又叫 Cupcake，这是第一个主要版本，发布于 2009 年 5 月。这个版本之后 Android 表现出来的能力才真正吸引了开发者的目光。用户界面得到了极大的改良。其主要的更新如下：拍摄/回放视频，并支持上传视频到 Youtube；支持立体声蓝牙耳机，同时改善自动配对性能；最新的采用 WebKit 技术的浏览器，支持复制/贴上和页面中搜索；GPS 性能大大提高；提供屏幕虚拟键盘；主屏幕增加音乐播放器和相框 widgets；应用程序自动随着手机旋转；短信、Gmail、日历、浏览器的用户接口大幅改进，如 Gmail 可以批量删除邮件；相机启动速度加快，拍摄图片可以直接上传到 Picasa；来电照片显示。

15.3.3 Android 1.6

搭载 Android 1.6 的 HTC Hero 获得了意想不到的成功。Android 最终获得了非 Android 用户的关注，并且也让竞争者苹果和微软公司更加警惕了。Android 1.6 Donut 2009 年 9 月发布。主要特性有：Android 应用市场集成；照相；摄像以及浏览；多选/删除功能；手势搜索；语音搜索应用集成；极大提升了语音阅读功能；对非标准分辨率有了更好的支持。

15.3.4 Android 2.0/2.1

Android 2.0.1 SDK 于 2009 年 12 月发布，开发代号为 Eclair，之后是 2010 年 1 月的 2.1 版本。主要更新如下：优化硬件速度；支持更多的屏幕分辨率；大幅度的用户界面改良；新的浏览器的用户接口和支持 HTML5；新的联系人名单；改进 Google Maps 3.1.2；支持内置相机闪光灯；改进的虚拟键盘；支持蓝牙 2.1；支持动态桌面的设计。

15.3.5 Android 2.2

该版本名字叫 Froyo,中文代号称“冻酸奶”,Nexus One 的用户已经开始接受 Android 2.2 Froyo 的无线更新,其他机型的用户应该会在短时间内获取这一更新。这一版本包含大量让其他手机用户垂涎三尺的更新。

其主要特性有:完整的 Flash 10.1 支持;最高 5 倍速度提升;最多支持 8 个设备连接的移动热点功能;有着专用链接的改进版主屏幕;大量 Exchange 支持改进,支持 Exchange 2010,包括远程数据抹除,自动发现服务,完整的日程表支持,全局联系人列表查找;摄像头/视频改进,比如更好的屏幕控制按钮;大众喜好的录像和开启闪光灯照明;多键盘语言支持;Android 云信息将应用安装在记忆卡上;蓝牙语音拨号;支持 720P 视频录制。

15.3.6 Android 2.3

2010 年 12 月,谷歌公司在发布旗下第二款手机 Nexus S 的同时,全新的谷歌 Android 2.3 Gingerbread(姜饼)系统正式亮相。相比 Android 2.2 版本来说,Android 2.3 Gingerbread 在系统性能方面的提升倒不是很大,更重要的是它让 Android 系统的功能更加全面。

其主要特性有:用户界面更美观;进一步提升运行速度;提升游戏体验;提升多媒体能力;增加官方进程管理;改善电源管理;NFC 近场通信;全局下载管理;改进的虚拟键盘;原生支持前置摄像头;SIP 网络电话。

15.3.7 Android 3.0/3.1

Android 3.0 命名为 Honeycomb,于 2011 年 2 月发布,该版本主要更新有:仅供平板电脑使用;Google eBooks 上提供数百万本书;支持平板电脑大屏幕、高分辨率;新版 Gmail;Google Talk 视讯功能;3D 加速处理;网页版 Market(Web store)详细分类显示,依个人分别设定安装应用程序;新的短消息通知功能;专为平板电脑设计的用户界面。

15.3.8 Android 4.0

Android 4.0 命名为 Ice Cream Sandwich(ICS,冰淇淋三明治),是谷歌公司发布的新一代操作系统。Android 4.0 拥有十大重点新特性,集成谷歌电视和 Chrome OS 的智能停放,运行速度得到进一步提升。2011 年 10 月,谷歌与三星联手在中国香港举办发布会,公布有关 Android 4.0 系统的信息,其主要更新有:统一的 UI 框架 & 沟通与共享;全新的媒体处理能力;全新的连接类型;全新的 UI 组件及功能;全新的输入方式及文本服务;应用程序及内容安全性;企业业务增强。

15.4 Android 系统架构

Android 和其他操作系统一样采用了分层的架构。Android 分为 4 层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层，Android 系统架构如图 15-4-1 所示。

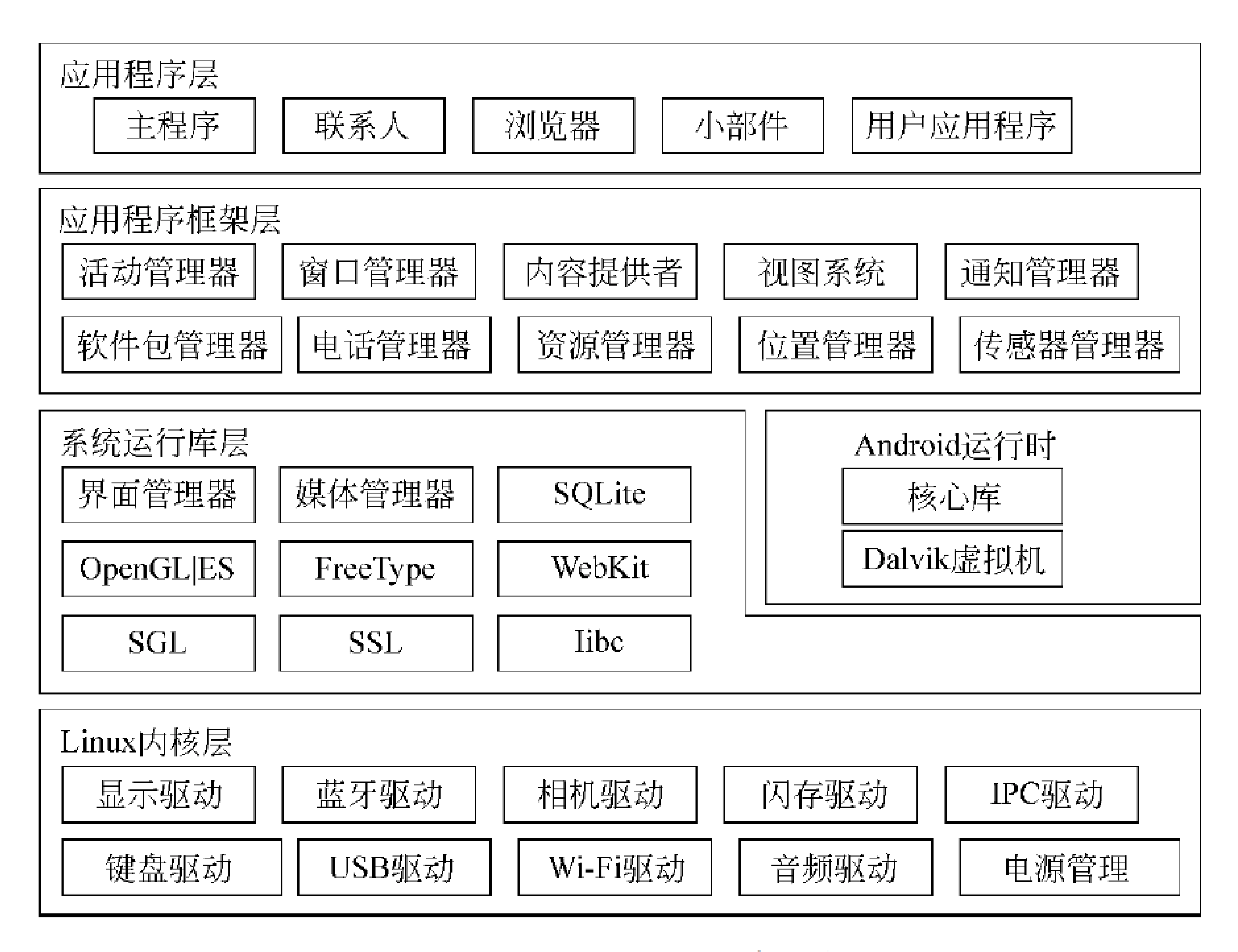


图 15-4-1　Android 系统架构

15.4.1 应用程序和小部件

购置 Android 手机时，手机会同一系列标准应用程序一起发布，这些应用程序包括 Email 客户端、SMS 短消息程序、日历、地图、浏览器和联系人管理程序等。所有的应用程序都是使用 Java 语言编写的。用户可以从网上下载新的应用程序并安装在手机上以扩展用户的应用。当然，也可以自己开发新的应用程序，这将是开发人员大显身手的地方。

15.4.2 应用程序框架

应用程序框架层是 Android 开发的基础，该层简化了组件的重用，开发人员可以直接使用其提供的组件来进行快速的应用程序开发，也可以通过继承而实现个性化的拓展。很多系统级应用程序也是通过这一层来实现其核心功能的，与自己开发的应用程序使用的 API 是完全相同的，如果需要，甚至可以让自己的应用程序代替标准的应用程序。应用程序框架提供一系列的服务和系统，具体如下所示。

(1) 视图系统。丰富而又可扩展的视图(view)可以用来构建应用程序视图，它包括

列表(list)、网格(grid)、文本框(text box)和按钮(button)等,视图甚至可嵌入到 Web 浏览器。

(2) 内容提供器(content providers)。使得应用程序可以访问另一个应用程序的数据(如联系人数据库),或者共享它们自己的数据,如联系人信息。

(3) 资源管理器(resource manager)。提供非代码资源的访问,如本地字符串、图形和布局文件(layout file)。

(4) 通知管理器(notification manager)。使得应用程序可以在状态栏中显示自定义的提示信息。像收到短信、临界状态报警、临近预约时间和异常入侵等事件都可以通过友好的方式通知用户。

(5) 活动管理器(activity manager)。用来管理应用程序生命周期并提供常用的导航回退功能。

15.4.3 Android 运行时

Android 运行时包括核心库和 Dalvik 虚拟机。该核心库提供 Java 编程语言核心库的大多数功能,虽然该核心库与 Java SE 和 Java ME 库是不同的,但它们之间还是有很大一部分是相同的。

Dalvik 虚拟机的可执行文件针对小内存使用做了优化,所有的类都经由 Java 编译器编译,然后通过 SDK 中的 dx 工具转化成.dex 格式由虚拟机执行。Dalvik 虚拟机依赖于 Linux 内核的一些功能,比如线程机制和底层内存管理机制。每一个 Android 应用程序都在它自己的进程中运行,都拥有一个独立的 Dalvik 虚拟机实例。Dalvik 被设计成一个设备可以同时高效地运行多个虚拟系统。

15.4.4 本机库

Android 包含一些 C/C++ 库,这些库能被 Android 系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。以下是一些核心库。

(1) 系统 C 库。一个从 BSD 继承来的标准 C 系统函数库(libc),它是专门为基于 embedded Linux 的设备定制的。

(2) 媒体库。该库支持多种常用的音频、视频格式回放和录制,同时支持静态图像文件。编码格式包括 MPEG4、H.264、MP3、AAC、AMR、JPG 和 PNG 等。

(3) 界面管理器(surface manager)。它并不将显示内容直接绘制到屏幕缓冲区,而是将绘制命令传递给屏幕外的位图,然后该位图与其他位图组合起来形成用户看到的显示内容。

(4) 浏览器引擎。为保证快速地显示 HTML 内容,Android 使用 WebKit 库支持在应用中嵌入 Web 视图。

(5) 2D 和 3D 图形。Android 可将 2D 和 3D 元素结合到一个用户界面,2D 和 3D 分别基于 SGL 和 OpenGL ES 实现。OpenGL ES 库可以使用硬件 3D 加速(如果可用)或者使用高度优化的 3D 软加速。

(6) FreeType 用于位图(bitmap)和矢量(vector)字体显示。

(7) SQLite 数据库。这是一个轻量级的功能强劲的轻型关系型数据库引擎,iPhone 等手机产品使用的也是这款数据库系统。

15.4.5 Linux 内核

Linux 是硬件和软件层之间的抽象层,Android 是基于 Linux 2.6 的内核,但是把 Linux 内受 GNU 协议约束的部分做了取代,这样 Android 的程序就可以用于商业目的。Android 核心系统服务如安全性、内存管理、进程管理、网络协议以及驱动模型都依赖于 Linux 内核。

Android 的 HAL(硬件抽象层)是能以封闭源码形式提供硬件驱动模块。HAL 的目的是为了把 Android framework 与 Linux 内核隔开,让 Android 不至过度依赖 Linux 内核,以达成 kernel independent 的概念,也让 Android framework 的开发能在不考虑驱动程序实现的前提下进行发展。

15.4.6 安全权限机制

Android 本身是一个权限分立的操作系统。在这类操作系统中,每个应用都以唯一的一个系统识别身份运行(Linux 用户 ID 与群组 ID)。系统的各部分也分别使用各自独立的识别方式。Linux 就是这样将应用与应用,应用与系统隔离开。系统更多的安全功能通过权限机制提供。权限可以限制某个特定进程的特定操作,也可以限制每个 URI 权限对特定数据段的访问。

Android 安全架构的核心设计思想是,在默认设置下,所有应用都没有权限对其他应用、系统或用户进行较大影响的操作。这其中包括读写用户隐私数据(联系人或电子邮件),读写其他应用文件,访问网络或阻止设备待机等。安装应用时,在检查程序签名提及的权限,且经过用户确认后,软件包安装器会给予应用权限。从用户角度看,一款 Android 应用通常会要求如下的权限:拨打电话,发送短信或彩信,修改/删除 SD 卡上的内容,读取联系人的信息,读取日程信息,写入日程数据,读取电话状态或识别码,精确的(基于 GPS)地理位置,模糊的(基于网络获取)地理位置,创建蓝牙连接,对互联网的完全访问,查看网络状态,查看 WiFi 状态,避免手机待机,修改系统全局设置,读取同步设定,开机自启动,重启其他应用,终止运行中的应用,设定偏好应用,震动控制,拍摄图片等。

一款应用应该根据自身提供的功能要求合理的权限。用户也可以分析一款应用所需的权限,从而简单判定这款应用是否安全。如一款应用是不带广告的单机版,也没有任何附加的内容需要下载,那么它要求访问网络的权限就比较可疑。

15.5 SL4A、PHP 与 Android

早期的 Android 应用程序开发,通常是通过 Android SDK(Android 软件开发包)使用 Java 作为编程语言来开发应用程序。能否使用功能强大的 PHP 脚本语言构建

Android 应用程序呢？回答是肯定的，但需要借助 SL4A 和 PFA 两个开源项目。

SL4A 的全称为 Scripting Layer for Android，之前称作 ASE，中文翻译为 Android 的脚本架构层，它的目的就是可以用熟知的脚本开发语言来开发 Android 应用程序。SL4A 支持的脚本语言包括 Python、Perl、JRuby、Lua、BeanShell、JavaScript、Tcl 和 shell 等。SL4A 工作原理基于 RPC 远程调用，通过本地的脚本解析器和远端的原生态 Android Server 层的 APK 进行信息交互，即实现一个远程代理，把本地脚本的函数调用通过 JSON 格式的封装，传递给远程原生态 Server APK 进行实际的 Android 系统函数呼叫，最后将操作的执行结果反馈给本地脚本解析器，然后再在终端显示出运行结果。所以任何本地脚本语言，只要实现了这套兼容的 JSON RPC 通信接口，就可以呼叫 SL4A 的 RPC Server 端程序。SL4A 的另一个工作原理是调用 Android 的 API。通过 SL4A 的接口，可以实现发送通话请求、发送文字信息、扫描条形码、获取 GPS 信息和使用语音服务。

把脚本引入到 Android 系统中，为 Android 上的自动化测试提供了便利。因为脚本编写相对简单，可以很方便地实现诸多功能：操作 Intetnt；启动 Activity；拨打电话；发送短信；扫描条形；获取当前地理位置和传感器数据；使用 Text-To-Speech（TTS）等。有关 SL4A 的其他说明，可以到官方网站 http://code. google. com/p/android-scripting/ 参看。

SL4A 适用于以下场景：RAD programming，RAD 即 Rapid Application Development（快速原型开发），进行工程项目的可行性分析；编写测试脚本，通常代码量都比较小，可方便实现；开发工具软件包，来实现一些快捷的功能。

开源公司 IronTec 推出了一个开源项目 PHP for Android（简写为 PFA），把 PHP 带到 Android 平台上。PFA 项目通过 SL4A 让 PHP 可以用来构建 Android 应用程序。PFA 项目还提供了 APK 工具和文档为 ASE（Android Scripting Environment）提供 PHP 支持。PFA 的官方网站为 http://www. phpforandroid. net。

15.6 开发环境

要使用 PHP 开发 Android 应用程序至少需要 JDK（Java 开发工具包环境）、Android SDK、SL4A 和 PFA 四类软件，JDK 和 Android SDK 需要安装在计算机上，而 SL4A 和 PFA 需要安装在 Android 手机上。软件的安装次序：先安装 JDK 到计算机上，然后安装 Android SDK 和创建 Android 模拟器，最后在模拟器上安装 SL4A 和 PFA 软件。

15.6.1 JDK

首先需要安装 Java 的 JDK，所有的 Android 开发工具都需要它，JDK 版本要 JDK5 或更高版本。值得注意的是，如果打算使用 Java 开发 Android 应用程序，只安装 JRE（Java 运行环境）还不够，还需要安装完整的开发包。可以从 Oracle 网站 http://goo. gl/

yhhaL 下载并安装 JDK SE 6.0。在 Windows 操作系统 DOS 窗口下输入 java -version，如果显示 1.6.x 或大于 1.6 的版本号则说明安装成功。Java 环境测试结果如图 15-6-1 所示。

```
C:\Users\Administrator>java -version
java version "1.6.0_10-rc"
Java(TM) SE Runtime Environment (build 1.6.0_10-rc-b28)
Java HotSpot(TM) Client VM (build 11.0-b15, mixed mode, sharing)
```

图 15-6-1　Java 环境测试

15.6.2　Android SDK 安装

可以通过离线的方式或在线的方式搭建开发环境，不管哪种方式，都需要先下载 Android SDK 主安装包。由于在线方式搭建开发环境耗时较久，所以本节介绍离线方式搭建开发环境。先从官方网站 http://dl.google.com/android/android- sdk_r22.2.1-windows.zip 下载较新的 Android SDK 主安装包，安装包含有 SDK 管理器、AVD 管理器和工具包。下载后直接解压缩到指定的文件夹即可，这里解压缩到 c:\android-sdk-windows 文件夹，解压后文件夹结构如图 15-6-2 所示。注意 SDK 不能解压缩到中文文件夹或者桌面上(文件夹名称中最好不要有空格)。刚解压缩后的 platforms 和 add-ons 文件夹是空的，解压缩后其占用空间大小为 135MB。

add-ons	2013/9/17 19:12	文件夹
platforms	2013/9/17 19:12	文件夹
tools	2013/9/17 19:12	文件夹
AVD Manager	2013/9/17 19:12	应用程序
SDK Manager	2013/9/17 19:12	应用程序
SDK Readme	2013/9/17 19:12	文本文档

图 15-6-2　Android SDK 目录结构

Android SDK 并没有提供开发环境中的一切，还缺少平台、平台工具和模拟器映像文件等软件。接下来依次完成这些内容。Platforms 文件夹下存放不同版本的 Android 系统，可从官方网站下载文件 https://dl-ssl.google.com/android/repository//android-14_r01.zip，下载后解压缩文件 android-14_r01.zip，把解压缩后的 android-4.0.1 文件夹复制到 c:\android-sdk-windows\platforms 文件夹中。

platform-tools 文件夹下存放了不同 Android 平台相关的工具，从官方网站下载 https://dl-ssl.google.com/android/repository/platform-tools_r14-windows.zip，下载后解压缩文件 platform-tools_r14-windows.zip，把解压缩后的 platform-tools 文件夹复制到 c:\android-sdk-windows 文件夹中。

build-tools 文件夹下存放了发布 Android 应用程序相关的工具，解压后的 Android SDK 并没有此文件夹，需要先手动在 c:\android-sdk-windows 文件夹中建立 build-tools 子文件夹，然后从官方网站下载 https://dl-ssl.google.com/android//repository//build-tools_r17-windows.zip，下载后解压缩 build-tools_r17-windows.zip 文件，最后把解压缩

后的 android-4.2.2 子文件夹复制到 c:\android-sdk-windows\build-tools 文件夹中。

system-images 文件夹存放了不同系统结构的 Android 映像文件，在 android-sdk-windows 文件夹里面先新建一个 system-images\android-14 子文件夹，然后解压缩下载的文件 sysimg_armv7a-14_r01.zip，最后把解压缩后的文件夹 armeabi-v7a 复制到 system-images\android-14 文件夹内即可。完成后的 android-sdk-windows 文件夹如图 15-6-3 所示。

AVD Manager	2013/9/17 19:12	应用程序
SDK Manager	2013/9/17 19:12	应用程序
SDK Readme	2013/9/17 19:12	文本文档
add-ons	2013/9/17 19:12	文件夹
build-tools	2014/5/26 0:07	文件夹
platforms	2014/5/25 23:37	文件夹
platform-tools	2014/5/26 16:32	文件夹
system-images	2014/5/25 23:38	文件夹
tools	2013/9/17 19:12	文件夹

图 15-6-3　解压缩后的 Android SDK 文件夹

解压缩上述文件后，还需要修改环境变量 Path，把 Android SDK 目录加进来。在环境变量 Path 内容的末尾加进如下内容：

```
c:\android-sdk-windows\tools;c:\android-sdk-windows\platform-tools
```

同时，还需要新建环境变量 ANDROID_SDK_HOME，其值为

```
c:\android-sdk-windows
```

15.6.3　创建模拟器

做完上述工作，就可以开发 Android 程序了。Android 应用程序需要运行在 Android 手机上，但如果没有 Android 手机，那么可以创建 Android 模拟器代替 Android 手机，当然，模拟器并不能完全模拟手机，比如手机信号的强度等。接下来创建 Android 模拟器。运行 c:\android-sdk-window\AVD Manager.exe 程序，弹出 Android 虚拟设备管理器对话框，单击 New 按钮进入创建模拟器界面，然后按图 15-6-4 填写参数，AVD Name 参数表示模拟器名称，可自行定义，此处是 MyG1；Target 参数表示所创建的模拟机所模拟的 Android 版本，此处选中 Android 4.0-API Level 14；SD Card 参数表示所模拟 SD 卡的容量大小，此处模拟的 SD 卡容量是 256MB；Device 参数表示手机屏幕设备，此处选 3.2" QVGA(ADP2)(320 * 480：mdpi)；Memory Options 参数表示手机内存容量，此处为 512MB。

单击 OK 按钮创建 Android 模拟器，创建模拟器后将回到 Android 虚拟设备管理器界面，选中刚才创建的 MyG1 设备，然后单击 Start 按钮，弹出 Launch 对话框，单击对话框的 Launch 按钮即可运行 Android 模拟器。运行结果如图 15-6-5 所示，单击正下角的按钮将看到内置的应用程序，直接单击程序图标就可以运行对应的程序了。

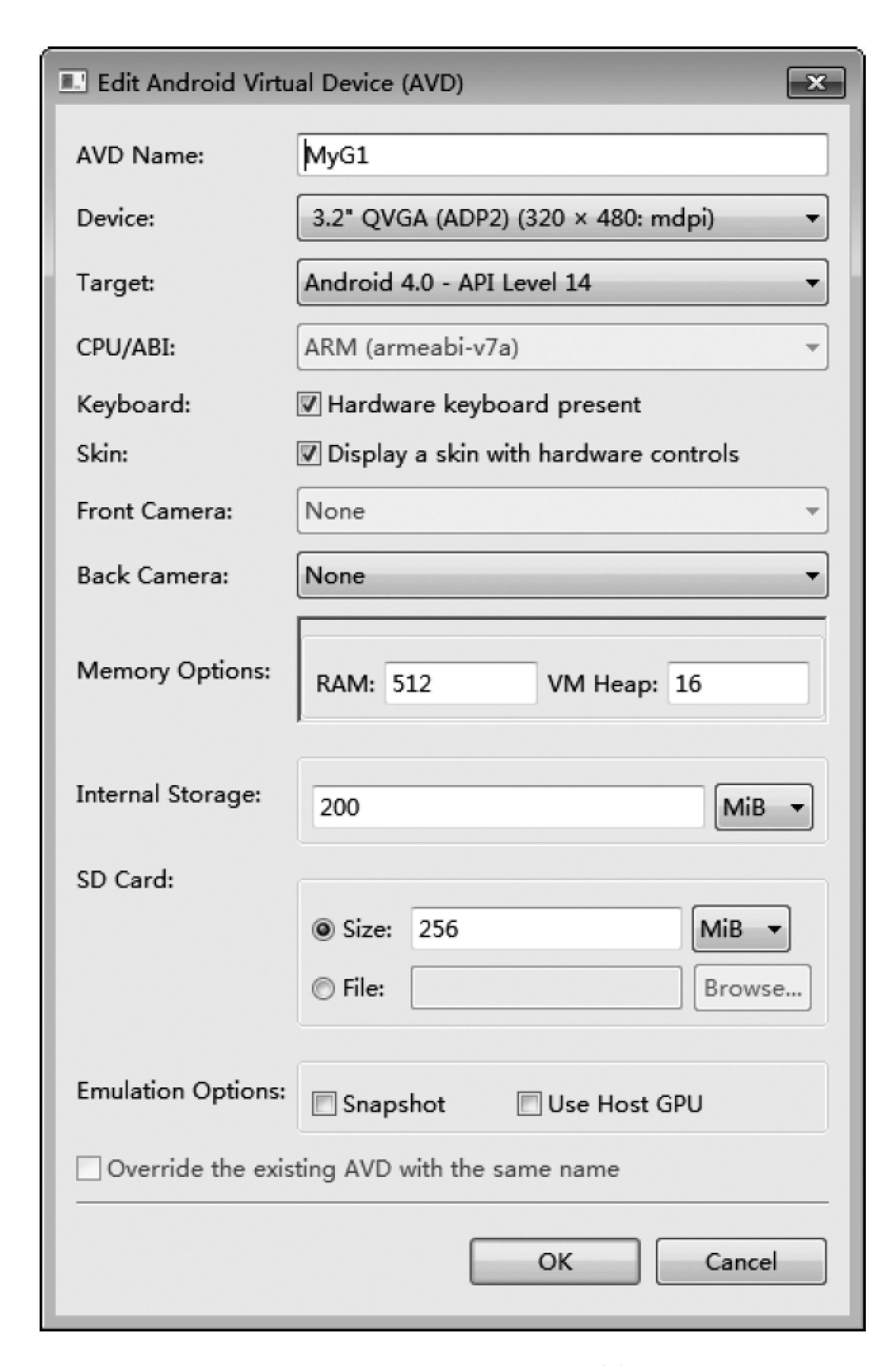


图 15-6-4 创建 Android 模拟器



图 15-6-5 Android 模拟器

15.6.4　SL4A 与 PFA

关于 SL4A，可以到网站 http://code.google.com/p/android-scripting/downloads/list/下载最新版 SL4A，目前最新版 SL4A 对应的文件名是 sl4a_r6.apk。在模拟器运行状态下，可在 Windows 的 DOS 命令窗口执行如下命令安装 SL4A：

```
adb install sl4a_r6.apk
```

如果 SL4A 成功安装，将显示图 15-6-6 的内容。

```
* daemon not running. starting it now *
* daemon started successfully *
848 KB/s (244019 bytes in 0.281s)
        pkg: /data/local/tmp/ASEr26unofficial.apk
Success
```

图 15-6-6　成功安装 SL4A

PHP for Android 软件可到官方网站 https://code.google.com/p/php-for-android/下载，其对应的文件名为 phpforandroid_r1.apk。在模拟器运行状态下，可在 Windows 的 DOS 窗口执行命令安装 phpforandroid_r1.apk，命令如下：

```
adb install phpforandroid.apk
```

如果 PHPFORANDROID 成功安装，将显示图 15-6-7 的内容。

```
440 KB/s (27935 bytes in 0.062s)
        pkg: /data/local/tmp/PhpForAndroid_r1.apk
Success
```

图 15-6-7　成功安装 PFA

安装 SL4A 和 PFA 后切换到模拟器，单击主屏幕最下方的按钮显示应用程序列表，可以看到 SL4A 和 PFA 图标，如图 15-6-8 所示，这表明两者都安装成功。

尽管在 DOS 窗口下已把 phpforandroid_r1.apk 安装到了模拟器，但还需要在模拟器中再执行 phpforandroid_r1.apk 进行第二次安装，第二次安装时，它将自动通过网上下载安装的文件并进行安装。值得注意的是，第二次安装时计算机要能上网，而且网络最好不要是局域网，否则还需要进行代理上网等设置。在模拟器中单击 PFA 图标进入安装界面并单击 Install 按钮，此时将从网上下载相应组件和进行第二次安装，下载和安装过程如图 15-6-9 所示。

下载安装完毕后，关闭 PFA，单击 SL4A 图标运行 SL4A 程序，将会看到一些 PHP 的实例程序，如图 15-6-10 所示。

15.6.5　运行第一个 Android 程序

单击其中任何一个程序都可以直接运行。单击 hello_world.php 程序图标，开始执行程序，弹出一个对话框，在其中输入一个字符串 tom，程序在屏幕中输出“Hello，tom”，运行结果如图 15-6-11 和图 15-6-12 所示。



图 15-6-8　SL4A 和 PFA 图标

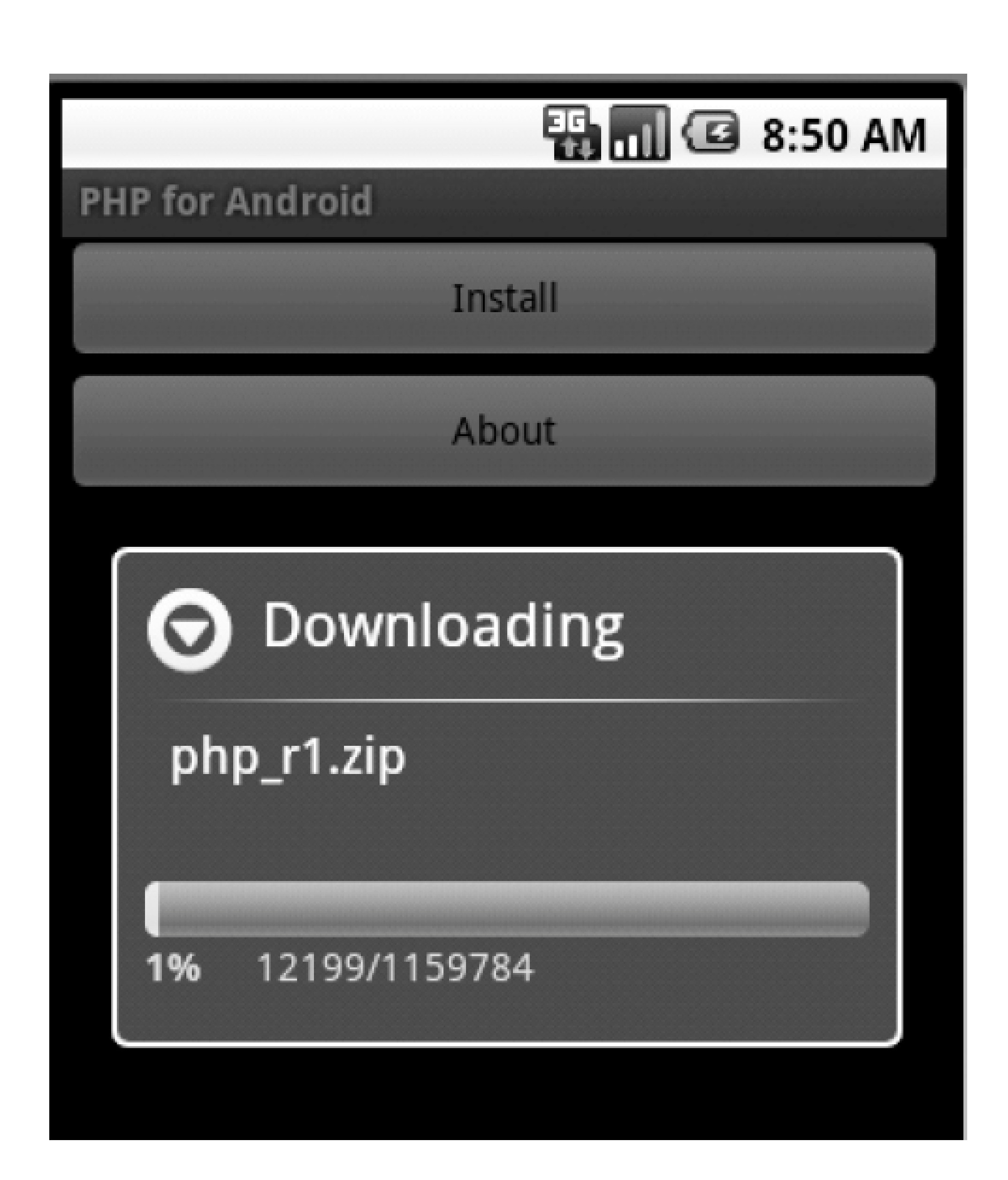


图 15-6-9　PFA 下载和安装组件

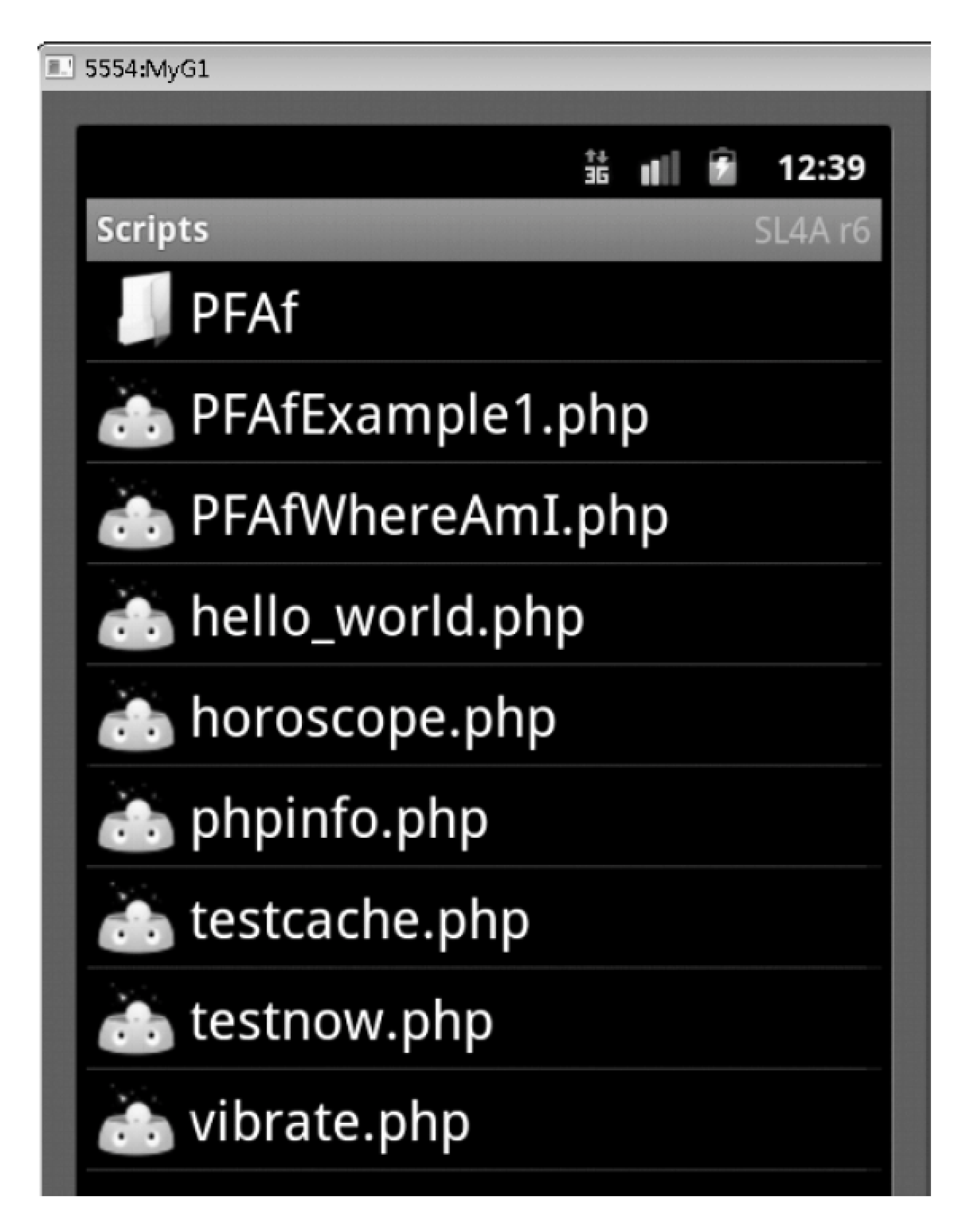


图 15-6-10　Android PHP 例子



图 15-6-11　对话框



图 15-6-12　程序的输出结果

第 16 章 chapter 16

Android 开发工具

16.1 工具简述

Android SDK 提供了很多比较常用的工具，这些工具对开发、调试和测试等工作都非常有用，它们分别位于 Android SDK 安装目录下的 platform-tools 和 tools 子目录中。下面先初步认识这些 Android 工具。

（1）zipalign 工具：根据数据结构对齐标准 DSA 对 APK 程序包进行对齐优化，从而提高 APK 应用程序与 Android 系统的交互效率。

（2）traceview 工具：traceview 是 Android 平台配备的性能分析工具，它可以通过图形化的方式跟踪程序的性能，并且能具体到方法。traceview 本身只是一个数据分析工具，而数据的采集则需要使用 Android SDK 中的 Debug 类或者利用 DDMS 工具。

（3）SQLite3 工具：SQLite3 数据库操作工具包括了对数据库表的增、删、改和查等操作，它可以运行在计算机上，也可以运行在 Android 手机端。SQLite3 库包含一个名为 sqlite3 的命令，用户可使用它手工输入并执行 SQL 命令管理数据。

（4）monkey 工具：压力测试工具，可在模拟器或设备的 adb shell 中向 Android 系统发出一系列伪随机的用户事件流。

（5）monkeyrunner 工具：Android 系统提供了一套 API，可自定义命令或事件流，并通过 monkeyrunner 工具执行自定义的事件流。

（6）mksdcard 工具：mksdcard 是创建 SDCard 的工具，启动模拟器时加载 SDCard 镜像可以模拟真实设备的 SD 卡，使用该命令的好处是可以在多个模拟器间共享 SD 卡。

（7）hprof-conv 工具：内存分析工具，显示详细的内存占用信息，指出可疑的内存泄漏对象。

（8）hierarchyviewer 工具：UI 层级查看工具。

（9）emulator 工具：Android Emulator 是一款功能非常齐全的模拟器，电话本、通话等功能都可正常使用（当然，不能真的从这里打电话），甚至其内置的浏览器和 Google Earth/Maps 都可以联网。用户可以使用键盘输入，鼠标单击模拟器按键输入，甚至还可以使用鼠标单击、拖动屏幕进行操纵。

（10）draw9patch 工具：draw9patch 是自动拉伸图片制作工具，它的一个优点是图片伸缩后不会发生变形，能较好地适用在不同尺寸的屏幕中。

（11）dmtracedump 工具：根据.trace 文件生成函数调用关系图。

（12）DDMS 工具：它是 Android IDE（集成开发环境）和模拟器以及真实设备之间的桥梁，开发人员可以通过 DDMS 看到在模拟器和真实设备上运行的进程状态，可以查看开发调试日记信息，可以向目标机发送短信和打电话，可以向模拟器和手机发送地理位置信息，除此，它还有很多强大的其他功能。

（13）apkbuilder 工具：生成 APK 文件的工具。

（14）avd 工具：avd 是 Android 虚拟设备管理器，用它来模拟手机设备，和 emulator 相同，它是 emulator 的快捷方式。

（15）adb 工具：Android 提供的一个通用调试工具，通过它可以管理和操作设备或手机模拟器的状态。

（16）aapt 工具：将源码、资源文件和配置文件打包成 APK 文件或解压 APK 文件。

（17）aidl 工具：根据.aidl 文件生成 AIDL 的 Java 实现，用 Eclipse 进行 AIDL 开发时，ADT 插件调用此命令自动将.aidl 文件生成 Java 实现。

（18）dx 工具：将 Java 字节码转换成 Android 字节码并写入.dex 文件中。

（19）dexdump 工具：APK 反编译工具。

16.2 文　　档

可以打开 Android SDK 安装目录下的 docs 目录下的 index.html 文件查看 Android SDK 文档，通过文档可以学习 Android 的使用。文档由以下几个部分组成。

（1）Home：欢迎界面。

（2）SDK：包含各版本 SDK 的概述、对开发环境的搭建的指导等一些开发准备知识。

（3）Dev Guide：对开发本身进行了比较初级的指导，一些比较实用的例子都在这里，同时，在这里还可以学习使用 Android 附带的 adb 等工具，包括本章介绍的工具。

（4）Reference：类似于 Java API 文档的 Android API 文档，如果用 Java 开发 Android 程序，那么可在这里查看说明书。

（5）Resources Video 和 Blog：对开发者的一些其他支持资源，包括视频教学资源和博客资源。

16.3 SDK 和 AVD Manager

开发人员可以使用 SDK 和 AVD Manager 下载模拟器和设置模拟器。关于 SDK 和 AVD Manager 可以参看第 15 章的内容。

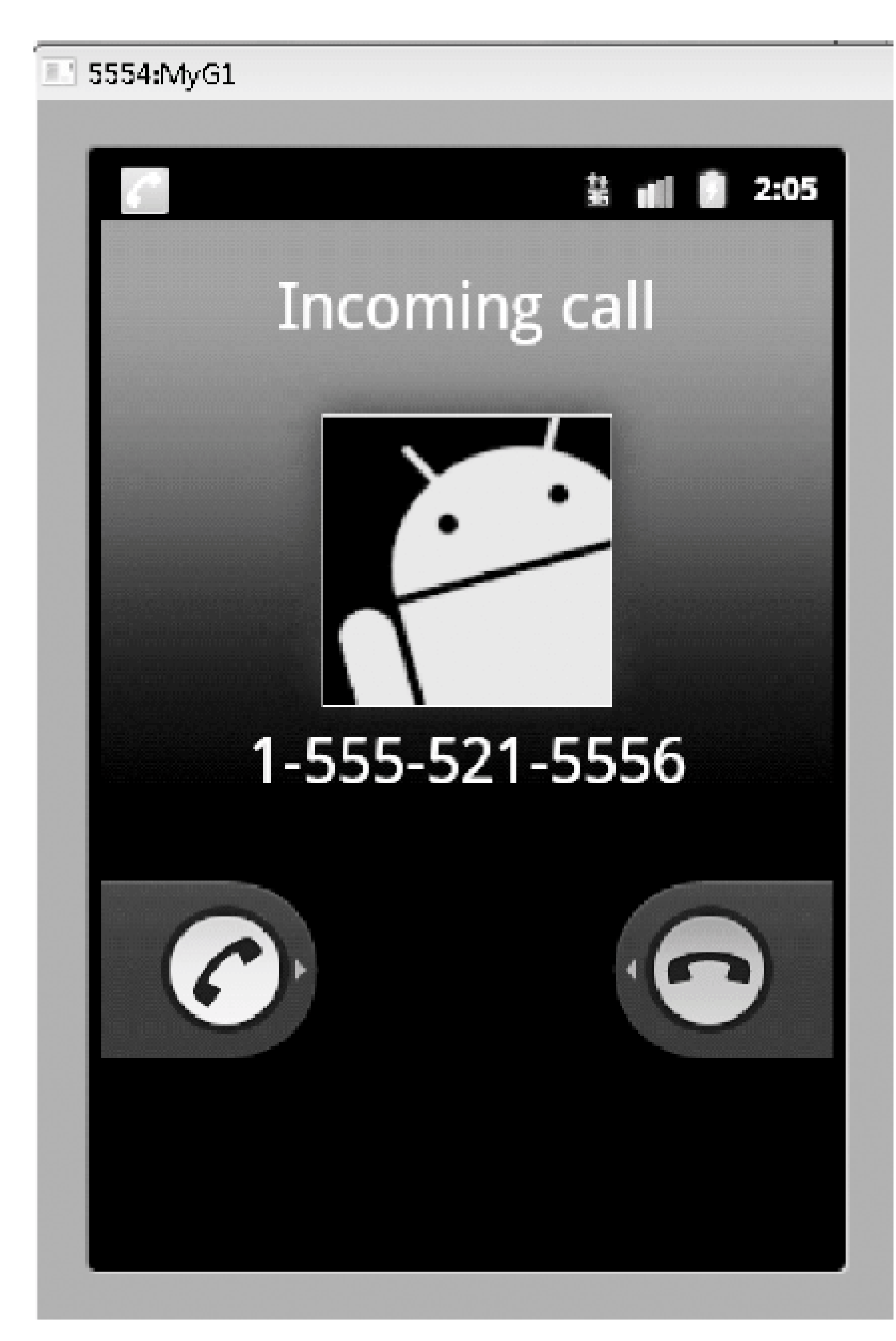
16.4 模 拟 器

16.4.1 模拟器给模拟器打电话

要模拟两部手机打电话，需要启动两个 Android 模拟器。首先通过程序 AVD Manager.exe 启动第一个模拟器，该程序在 Android SDK 安装目录下。启动后模拟器的 ID 通常为 5554，显示在模拟器窗口左上角。然后启动第二个模拟器，第二个模拟器的启动操作和第一个模拟器相同，通常 ID 为 5556。现在有了两个模拟器的 ID，这里的 ID 号相当于真机上的手机号码，收发短信和接打电话全靠它们。接下来测试接打电话和收发短信。怎么测试呢？很简单，就像用户有两部手机一样，A 手机跟 B 手机打，或者反过来。使用 5556 模拟器给 5554 模拟器拨打电话，先在 5556 模拟器的程序列表中找到 Phone 图标并单击运行电话程序，然后输入 5554，单击拨打图标开始拨打电话，最后两个模拟器都会跳出电话窗口，一个显示正在拨号，另一个提示有来电，其运行结果如图 16-4-1 所示，图 16-4-1(a)是 5556 模拟器在拨打电话，图 16-4-1(b)是 5554 模拟器提示有来电。



(a) 拨打电话



(b) 来电提示

图 16-4-1 两个模拟器通电话

16.4.2 模拟器给模拟器发短信

先启动两个模拟器，接下来使用 5556 模拟器给 5554 模拟器发短信。在 5556 模拟器程序列表中找到 Messaging 图标并单击运行，运行后选择 New message 菜单命令进入编写短消息界面，填写短消息内容“hello world!”和输入发送的号码 5554，单击 Send 按钮发送短消息，在 5554 模拟器的 Messaging 程序查看刚刚发送来的短消息内容。运行结

果如图16-4-2和图16-4-3所示，前者发短消息，后者接收短消息。



图16-4-2 发短消息

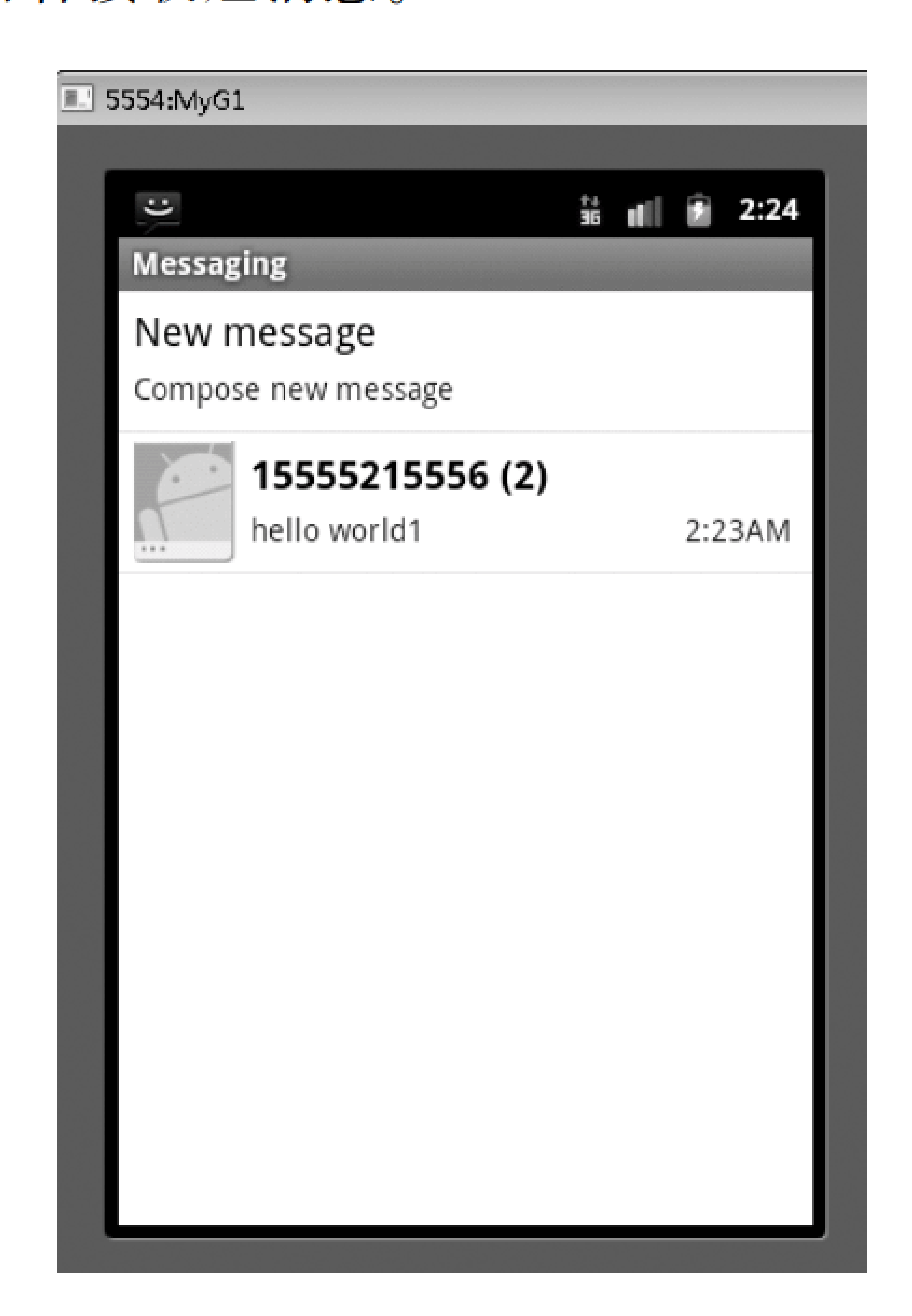


图16-4-3 收短信

注意：如果发现打不通或者发不了短信，查看一下当前模拟器的网络信号状态，如果是飞行模式，那么这项功能是实现不了的。

16.5 DDMS工具

DDMS的全称是Dalvik Debug Monitor Service，它提供了很多开发调试功能，包括为测试设备截屏、针对特定的进程查看正在运行的线程以及堆信息、logcat、广播状态信息、模拟电话呼叫、接收SMS和虚拟地理坐标等。开发人员可以通过DDMS看到目标机器上运行的进程状态，可以看进程的堆信息，可以查看logcat信息，可以查看进程分配内存情况，可以向目标机发送短信以及打电话，可以向Android模拟器发送地理位置信息，可以像GDB一样，attach某一个进程调试。在Android SDK安装目录下的tools子目录下找到ddms.bat批处理文件，双击运行该批处理文件即运行DDMS程序，在运行该程序之前先运行模拟器。DDMS运行界面如图16-5-1所示。

16.5.1 查看模拟器和程序信息

DDMS主界面的左上部列出了正在运行的模拟器名字、进程PID和程序包名等信息。图16-5-2列出了两个模拟器的名称、进程PID和程序软件包。

16.5.2 触发垃圾回收

在Android的开发中，要时刻关注内存的分配和垃圾回收，因为系统为每一个dalvik

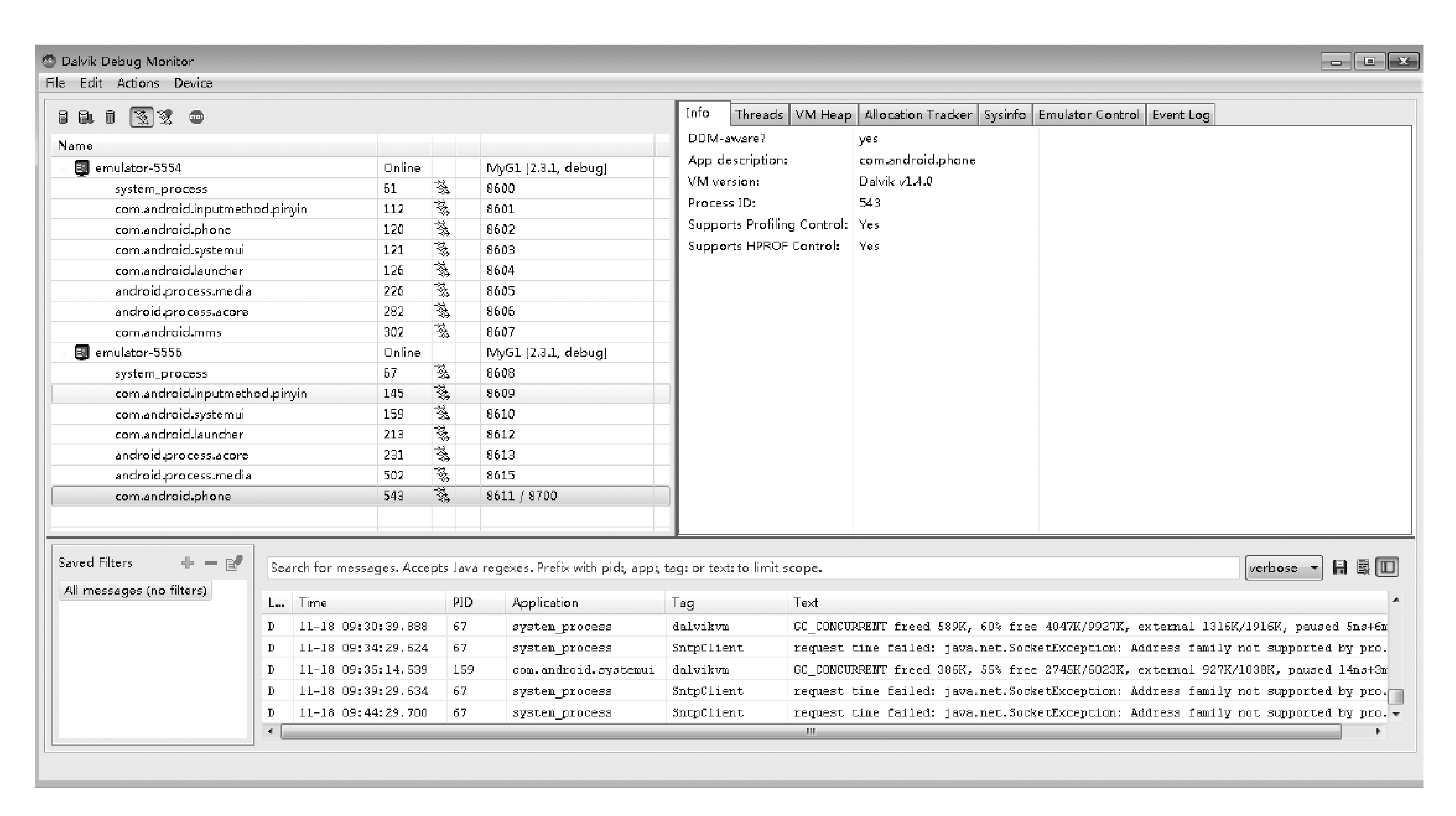


图 16-5-1 DDMS 主界面

Name			
emulator-5554	Online		MyG1 [2.3.1, debug]
system_process	61		8600
com.android.inputmethod.pinyin	112		8601
com.android.phone	120		8602
com.android.systemui	121		8603
com.android.launcher	126		8604
android.process.media	226		8605
android.process.acore	282		8606
com.android.mms	302		8607
emulator-5556	Online		MyG1 [2.3.1, debug]
system_process	67		8608
com.android.inputmethod.pinyin	145		8609
com.android.systemui	159		8610
com.android.launcher	213		8612
android.process.acore	231		8613
android.process.media	502		8615
com.android.phone	543		8611 / 8700

图 16-5-2 模拟器和运行程序信息

虚拟机分配的内存是有限的,在 Google 的 G1 中,分配的最大堆只有 16MB,小容量的存储空间清理垃圾对提高性能是必要的。Android 系统运行较长时间会生产出垃圾,系统会不定时地清理垃圾,也可以使用 DDMS 来强制进行垃圾回收(Garbage Collection,GC),步骤如下:

(1) 在模拟器或设备上,确认要进行垃圾回收的应用程序处于运行状态。

(2) 在 DDMS 中,找到这个应用程序的包,并且单击它使其高亮。

(3) 选择 Actions→Cause GC 菜单命令。

16.5.3 浏览模拟器文件

可以使用 DDMS 来查看并操作模拟器或设备上的 Android 文件系统。浏览 Android 文件系统的步骤如下：

(1) 在 DDMS 中，选择想要浏览的模拟器或设备。

(2) 选择 Device→File Explorer 菜单命令，系统将跳出目录浏览窗口，窗口列出了文件和目录基本信息，如图 16-5-3 所示。

(3) 单击某个目录项浏览该目录下的子文件夹或文件。

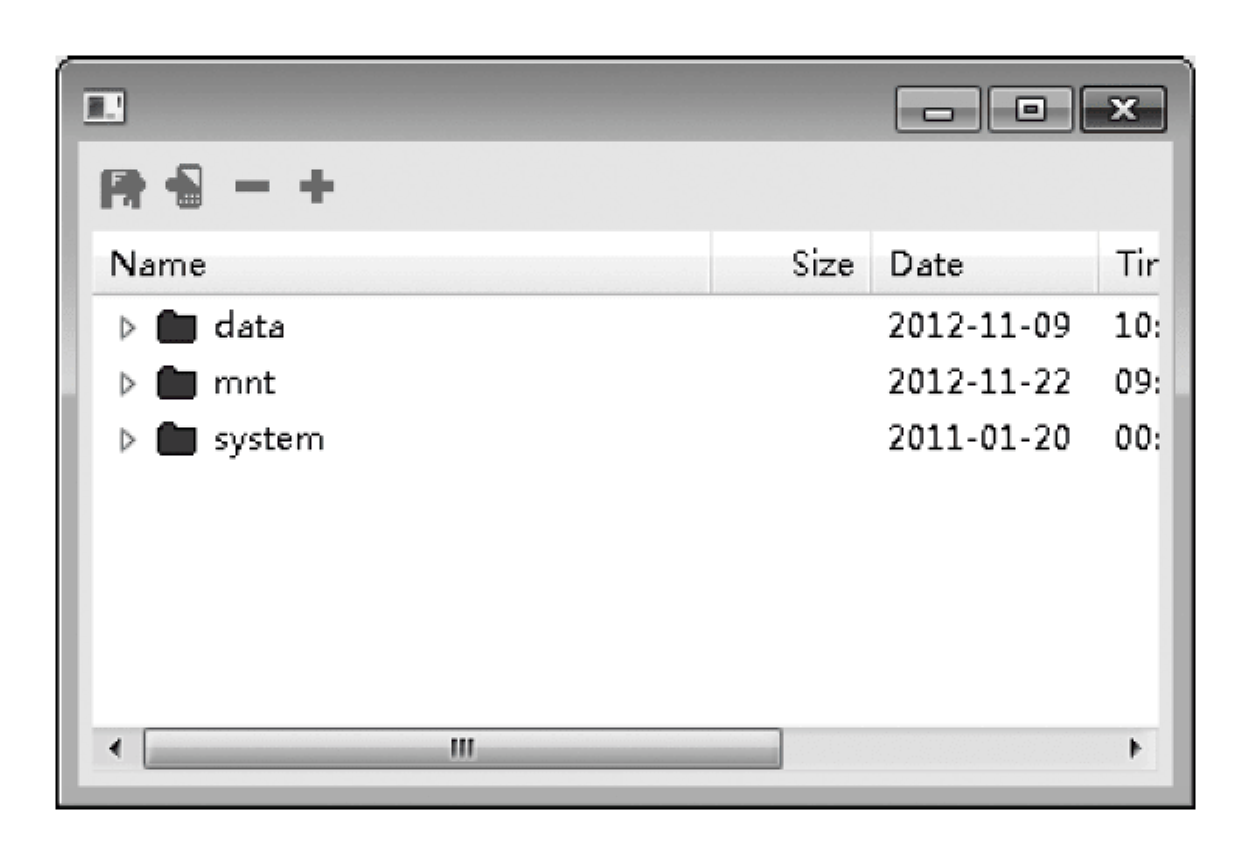


图 16-5-3　浏览文件和目录

16.5.4 向模拟器或设备复制文件

可以使用文件夹浏览器窗口将计算机上的文件复制到模拟器或设备的文件系统中，步骤如下：

(1) 使用文件夹浏览器导航至需要复制文件的文件夹，单击使其高亮。

(2) 在文件夹浏览器窗口的左上角，单击图标向设备中添加文件。

(3) 选择计算机上待复制的文件，然后选择 Open 菜单命令。

注意：文件夹浏览器还支持鼠标拖曳。这也是唯一可以向 Andriod 文件系统中复制文件夹的操作。不过，并不推荐向 Android 文件系统中复制文件夹，因为并没有用于删除它们的选项。但如果拥有许可权限，则需要使用程序来删除这些文件夹。总之，可以从计算机上将一个文件或文件夹拖到文件夹浏览器中，并在适当的位置释放它。

16.5.5 从模拟器或设备复制文件

可以使用文件夹浏览器窗口将模拟器或设备上的文件复制到计算机的文件系统中，步骤如下：

(1) 使用文件夹浏览器导航至需要复制文件的文件夹，单击使其高亮。

(2) 在文件夹浏览器窗口的左上角，单击图标把设备中的文件复制出来。

(3) 选择计算机保存目录和输入文件名，然后选择 Save 菜单命令。

16.5.6 删除和新建模拟器或设备上的文件夹

可以使用文件夹浏览器来删除模拟器或设备上的文件(但不能删除文件夹),步骤如下:

(1) 使用文件夹浏览器导航至需要删除的文件,单击使其高亮。

(2) 在文件夹浏览器的右上角,单击红色的减号图标➖来删除文件。

警示:执行这一操作时需要特别小心,因为没有任何确认提示,文件将立即删除并且没有办法恢复。如果要在模拟器或设备上新建文件夹,先选中父目录,然后单击文件夹浏览器窗口左上角的➕按钮,系统跳出对话框要求输入文件夹名称,在其中输入文件夹名并单击"确定"按钮即可。

16.5.7 打电话与发短信

要使用模拟器模拟语音呼入,需要执行以下步骤:

(1) 在 DDMS 主界面左侧区域中,选择想要拨打的模拟器。

(2) 单击 DDMS 主界面右侧区域中的 Emulator Control 标签进入语音呼入和短信发送界面。

(3) 在 Incoming number 文本框中输入模拟呼入的电话号码 5554,该号为模拟器的手机号码,它显示在模拟器窗口的左上角。

(4) 选择 Voice 单选按钮表示选择语音呼入功能。

(5) 单击 Call 按钮开始拨打电话,模拟器将会接收到呼入并响铃,如图 16-5-4 和图 16-5-5 所示。

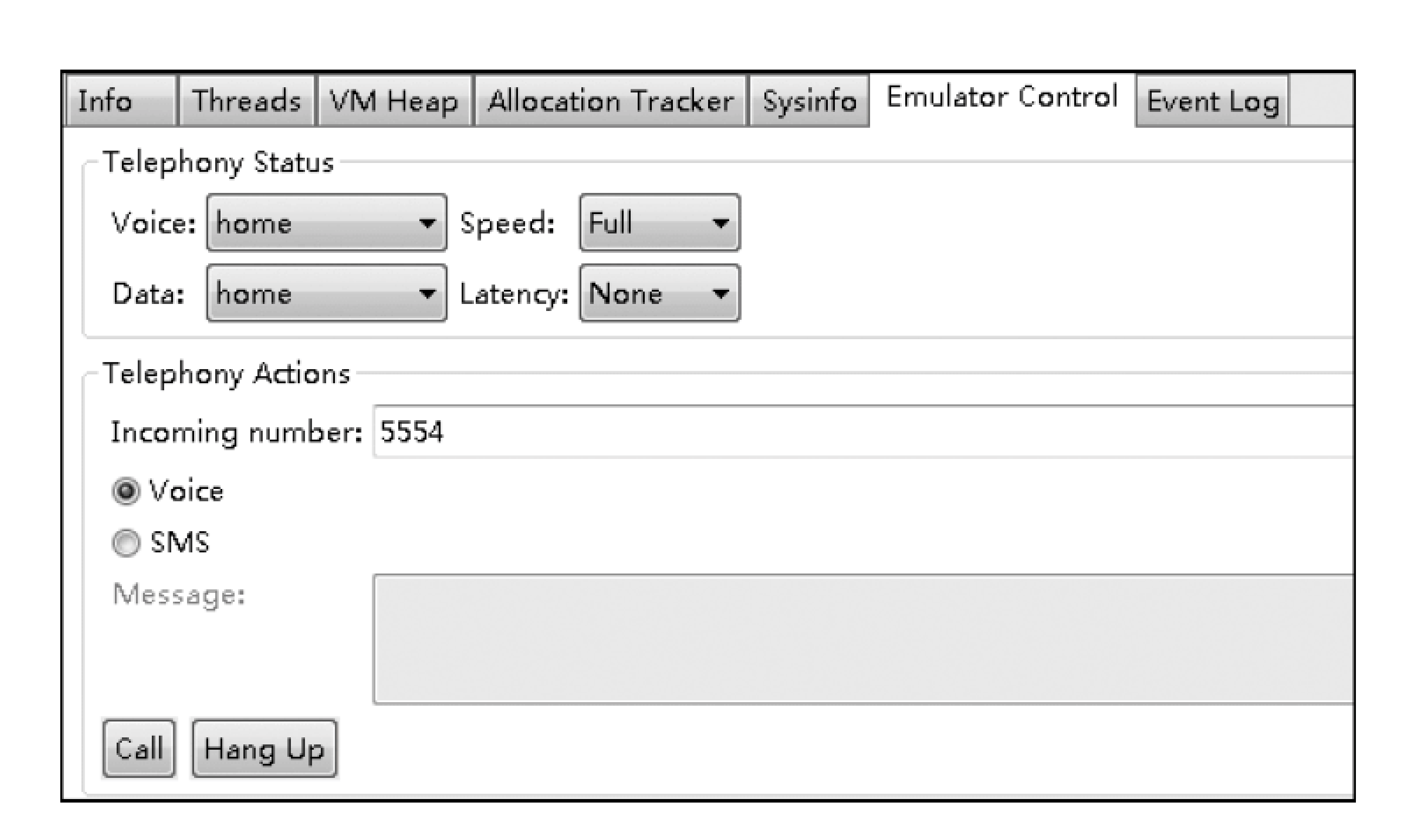




图 16-5-4 DDMS 拨打电话

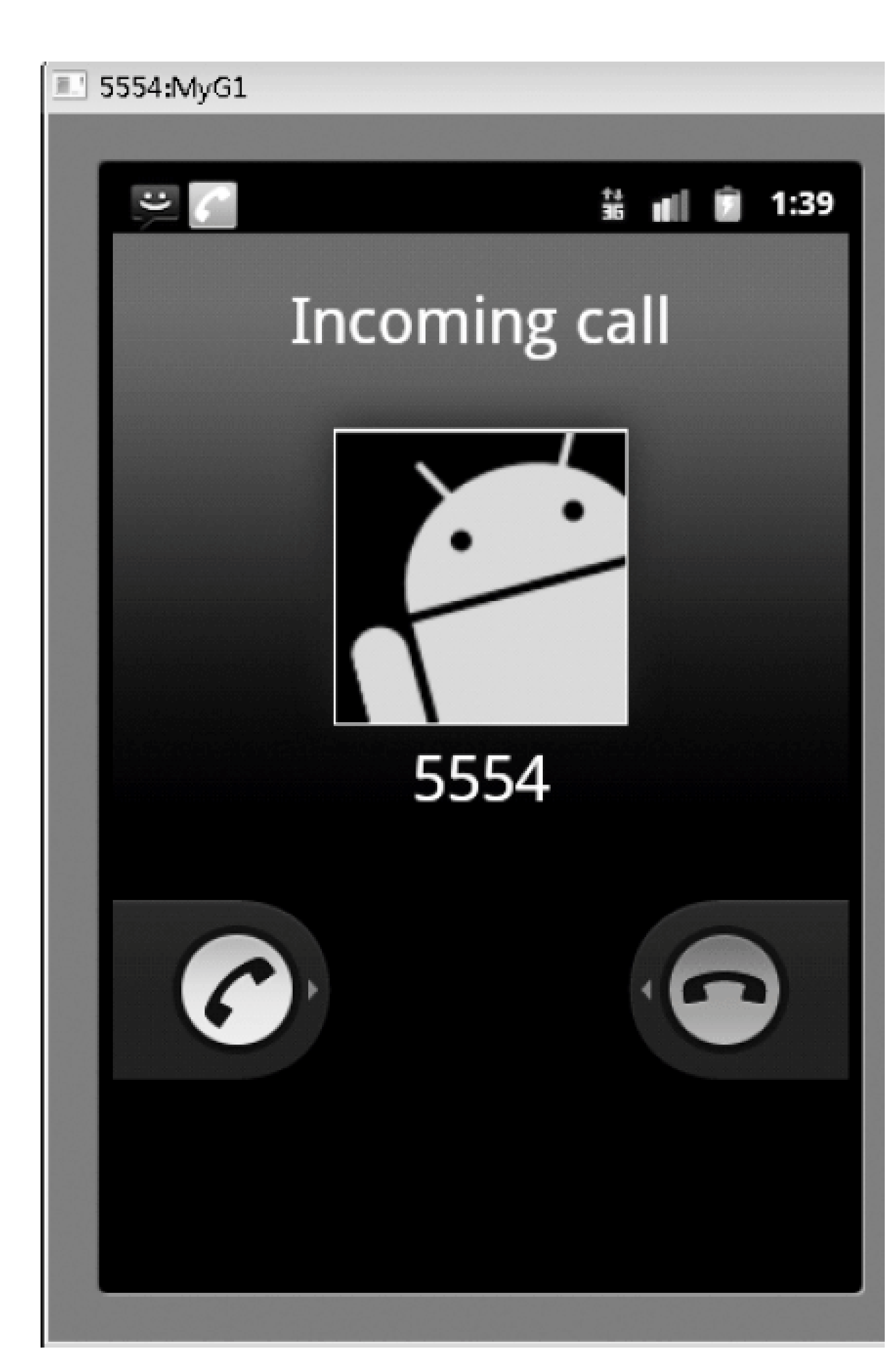


图 16-5-5 模拟器来电显示

模拟器可以像正常情况一样挂断电话，也可以使用DDMS中的Hang Up按钮终止通话。

DDMS提供了向模拟器发送SMS的最稳定的方法。其过程同模拟语音来电类似。要使用模拟器控制标签模拟发送SMS，步骤如下：

（1）在DDMS中，选择需要接收SMS的模拟器。

（2）单击Emulator Control标签进入语音呼叫和发短信界面。

（3）在Incoming number文本框中输入模拟发送电话号码5554。

（4）选择SMS菜单命令，发现Message文本框不再是灰色，表示可以在里面输入内容。

（5）在Message文本框中输入SMS消息的正文。

（6）单击Send按钮把短信发送给模拟器，模拟器会接收到SMS并显示通知。

可打开模拟器中的程序Messaging查看该短信。

16.5.8 拍摄模拟器屏幕截图

当需要抓取手机界面时，可以选择Device→Screen Capture菜单命令，系统会跳出一个窗口，该窗口会显示当时捕获的模拟器界面，该窗口还提供了刷新和保存图像等功能。

16.6 adb调试桥

adb的全称为Android debug bridhge，翻译成中文就是安卓调试桥程序。借助adb工具，可以管理设备或手机模拟器的状态，还可以进行很多手机操作，如安装软件、系统升级、运行shell命令和在手机模拟器上复制或粘贴文件等。简而言之，adb就是连接Android手机与PC端的桥梁，可以让用户在计算机上对手机或模拟器进行全面的操作。adb主要包括以下3个部分。

（1）adb client。运行在PC上，可以从控制台上运行adb命令启动adb client程序。

（2）adb daemon（守护进程）。运行于模拟器或手机。

（3）adb server（服务进程）。运行在PC上，管理adb client和adb daemon的通信，server与client通信的端口是5037。adb server与一个模拟器交互使用的端口有两个：一个是偶数端口（例如5554），专门用于与模拟器实例的连接，数据可以通过这个端口从模拟器转发给控制台；另一个是奇数端口（例如5555），专门与adb daemon连接供后面调试使用。如果有多个模拟器运行，则需要多个配对的奇偶端口，其端口范围是5555～5585。adb server通过扫描5555～5585的奇数端口来寻找模拟器或设备实例并与找到的实例建立链接。

16.6.1 安装和卸载应用

假设要安装一个book.apk文件，可以使用如下命令：

```
adb install book.apk
```

假设 ebook.apk 的包名是 net.xxy.book，可以使用如下命令卸载这个应用程序：

```
adb uninstall net.xxy.book
```

包是 Android 应用程序的唯一标识。如果在安装程序之前，该程序已经在模拟器或真机上存在了，需要先使用上面的命令卸载这个应用程序，然后再安装。或使用下面的命令重新安装，参数 r 表示重新安装。

```
adb install -r book.apk
```

在卸载应用程序时可以加上命令行参数 k 保留数据和缓冲目录，只卸载应用程序。命令如下：

```
adb uninstall -k net.xxy.book
```

16.6.2 进入设备或模拟器的 shell

Android 系统是基于 Linux 操作系统的，进入 Linux 的控制台可以控制整个系统。可以在 Windows 控制台输入如下命令进入模拟器的 shell：

```
adb shell
```

通过上面的命令进入设备或模拟器的 shell 环境后，可以执行各种 Linux 的命令，例如可以执行 ls 和 cd 命令查看文件列表和进入指定目录。

16.6.3 复制文件

用户可向一个设备或从一个设备中复制文件。如果要复制一个文件到设备或模拟器上，可使用 adb push 命令来执行。例如，要把 PC 上的文件 test.txt 复制到模拟器的/tmp/目录中，可以执行如下命令：

```
adb push test.txt /tmp/test.txt
```

从设备或模拟器上复制出一个文件可通过命令 adb pull 来完成。例如，要把模拟器上的文件/addroid/lib/libwebcore.so 复制到 PC 端当前目录，可输入以下命令：

```
adb pull /addroid/lib/libwebcore.so
```

16.6.4 启动和关闭 adb 服务

模拟器在运行一段时间后，adb 服务有可能会出现异常。这时需要关闭 adb 服务再重启该服务。关闭 adb 服务可使用如下命令来实现：

```
adb kill-server
```

在关闭 adb 服务后，可使用如下命令启动 adb 服务：

```
adb start-server
```

16.6.5 显示当前运行的全部模拟器

可以通过 adb devices 命令查看当前运行的全部模拟器，模拟器用"emulator-端口号"表示序列号，序列号用来唯一标识模拟器实例，运行结果如图 16-6-1 所示。

```
C:\Users\Administrator>adb devices
List of devices attached
emulator-5554   device
```

图 16-6-1 当前运行的全部模拟器

16.6.6 在指定的模拟器上执行命令

如果 PC 上运行多个模拟器，可以使用下面的命令选择某个模拟器来执行命令：

```
adb -s 模拟器名 shell
```

例如，执行下面的命令可以进入序列号为 emulator-5554 的模拟器的 Linux 控制台：

```
adb -s emulator-5554 shell
```

16.6.7 显示帮助信息

如果想了解 adb 的使用，可以在 DOS 窗口执行 adb help 命令查看 adb 帮助说明。

16.6.8 Android 模拟器间和模拟器与 PC 间通信

有时候开发网络游戏需要两机通信交换数据，如果使用真机调试比较麻烦，用两个模拟器进行通信会比较方便。下面先介绍模拟器与 PC 间的通信，再介绍模拟器间的通信。

两机通信需要 TCP/IP 协议的支持，可以在 DOS 命令行窗口输入如下命令查看模拟器的 IP，执行结果如图 16-6-2 所示。

```
adb -s emulator-5554 shell
getprop
```

```
C:\Users\Administrator>adb -s emulator-5554 shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
# getprop
getprop
[ro.secure]: [0]
[ro.allow.mock.location]: [1]
[ro.debuggable]: [1]
```

图 16-6-2 执行 getprop 命令

找到下面两行：

```
[net.eth0.dns1]:[10.0.2.3]
```

```
[net.gprs.local- ip]:[10.0.2.15]
```

第一行表示模拟器的 IP 为 10.0.2.15，第二行表示 DNS 为 10.0.2.3。Android 系统为实现模拟器间的通信会自动将 PC 的 IP 设置为 10.0.2.2（等同于 PC 本机的 IP 地址 127.0.0.1），用户可以在 Android 系统控制台和 PC 端 DOS 控制台窗口互相 ping 对方的 IP，会发现 Android 系统可以通过 IP（10.0.2.2）单向访问 PC，其结果如图 16-6-3 所示，而 PC 不能通过 IP 来直接访问 Android 系统。

```
C:\Users\Administrator>adb shell
# ping 10.0.2.2
ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
64 bytes from 10.0.2.2: icmp_seq=1 ttl=255 time=0.121 ms
64 bytes from 10.0.2.2: icmp_seq=2 ttl=255 time=0.182 ms
```

图 16-6-3　在 Android 系统下 ping PC

可以在 Android 系统设置端口重定向，使 PC 能访问 Android 模拟器。端口重定向需要 Telnet 客户端程序。如果没有安装，可以在 Windows 程序管理中的打开或关闭系统功能下找到 Telnet 客户端菜单项来启用 Telnet 客户端功能。

PC 和 Android 模拟器系统通信需要进行如下工作：

（1）运行模拟器。

（2）打开 Windows 控制台，执行如下命令：

```
telnet localhost 5554
```

5554 是模拟器的端口（位于 Android 模拟器窗口标题栏），执行之后会进入 Android 控制台，要注意的是，这个控制台不是 PC 的控制台，也不是 Android 操作系统的控制台，而是 Android 模拟器的控制台。Android 控制台界面如图 16-6-4 所示。

```
Android Console: type 'help' for a list of commands
OK
```

图 16-6-4　Android 控制台

（3）在 Android 控制台执行 redir 命令设置端口重定向，命令格式如下：

```
redir add <udp/tcp>:<PC 端口>:<模拟器端口>
```

执行下面命令之后，会把 PC 3000 端口接收到的 tcp/udp 数据转到模拟器的 3000 端口：

```
redir add udp: 3000: 3000
redir add tcp: 3000: 3000
```

多个 Android 模拟器通信需要进行如下工作：

（1）启动两个模拟器，通常端口为 5554 和 5556。

（2）打开 DOS 窗口，执行 telnet localhost 5554 命令连接到模拟器 5554。

（3）成功连接后，继续执行 redir add tcp:5000:6000 将 PC 端口 5000 绑定到模拟器 5554 的端口 6000 上。

(4) 此时模拟器 5556 通过向 PC 端口 5000(即地址 10.0.2.2:5000)发送 tcp/udp 数据包与模拟器 5554 通信。

(5) 打开 DOS 窗口,执行 telnet localhost 5556 命令连接到模拟器 5556。

(6) 成功连接后,继续执行 redir add tcp:5001:6001 将 PC 端口 5001 绑定到模拟器 5556 的端口 6001 上。

(7) 此时模拟器 5554 通过向 PC 端口 5001(即地址 10.0.2.2:5001)发送 tcp/udp 数据包与模拟器 5556 通信。

添加成功后,可以用 redir list 命令列出已经添加的映射端口,可以用 redir del 命令删除映射端口。

16.7 Hierarchy Viewer 工具

Hierarchy Viewer 是随 Android SDK 发布的非常有用而且使用简单的工具,位置在 tools 文件夹下,程序文件名为 hierarchyviewer.bat,它可以很方便地在开发者设计、调试和调整界面时提高用户的开发效率。Hierarchy Viewer 有以下两个主要功能:

(1) Load View Hierarchy。从可视化的角度直观地获得 UI 布局设计结构和各种属性的信息,帮助开发人员优化布局设计。

(2) Inspect Screenshot。进入界面精确查看模式。结合 debug 帮助观察特定的 UI 对象进行 invalidate 和 requestLayout 操作过程。

接下来看如何在 Load View Hierachy 中查看界面中各个控件的层次结构关系。先运行模拟器中的 PFA 程序,其界面如图 16-7-1 所示。



图 16-7-1 PFA 程序界面

接着双击 Android sdk 安装目录下的 tools 子目录下的 hierarchyviewer.bat 文件,Hierarchy Viewer 程序会列出模拟器和模拟器程序的包名,加粗部分是模拟器正在运行的 PFA 程序,如图 16-7-2 所示。

单击 Load View Hierarchy 按钮,由于要解析程序的相关窗口,所以这个过程要几秒钟,解析工作完成后会显示程序界面布局结构和程序控件属性等信息,如图 16-7-3 所示。解析后也可以单击界面左下角的按钮进入 Load View Hierarchy 界面。

从图 16-7-3 可知,Load View Hierachy 视图被划分为 4 个部分。最左边部分(面积最大一块)显示界面控件的层次结构,称为主窗口。右上方部分以缩略图的方式显示整个应用中各控件的层次关系,当一个界面中的控件比较多的时候,可以通过鼠标在这个显示区域进行移动,左边的主窗口中会具体显示相关的控件信息。右边区域的中间部分

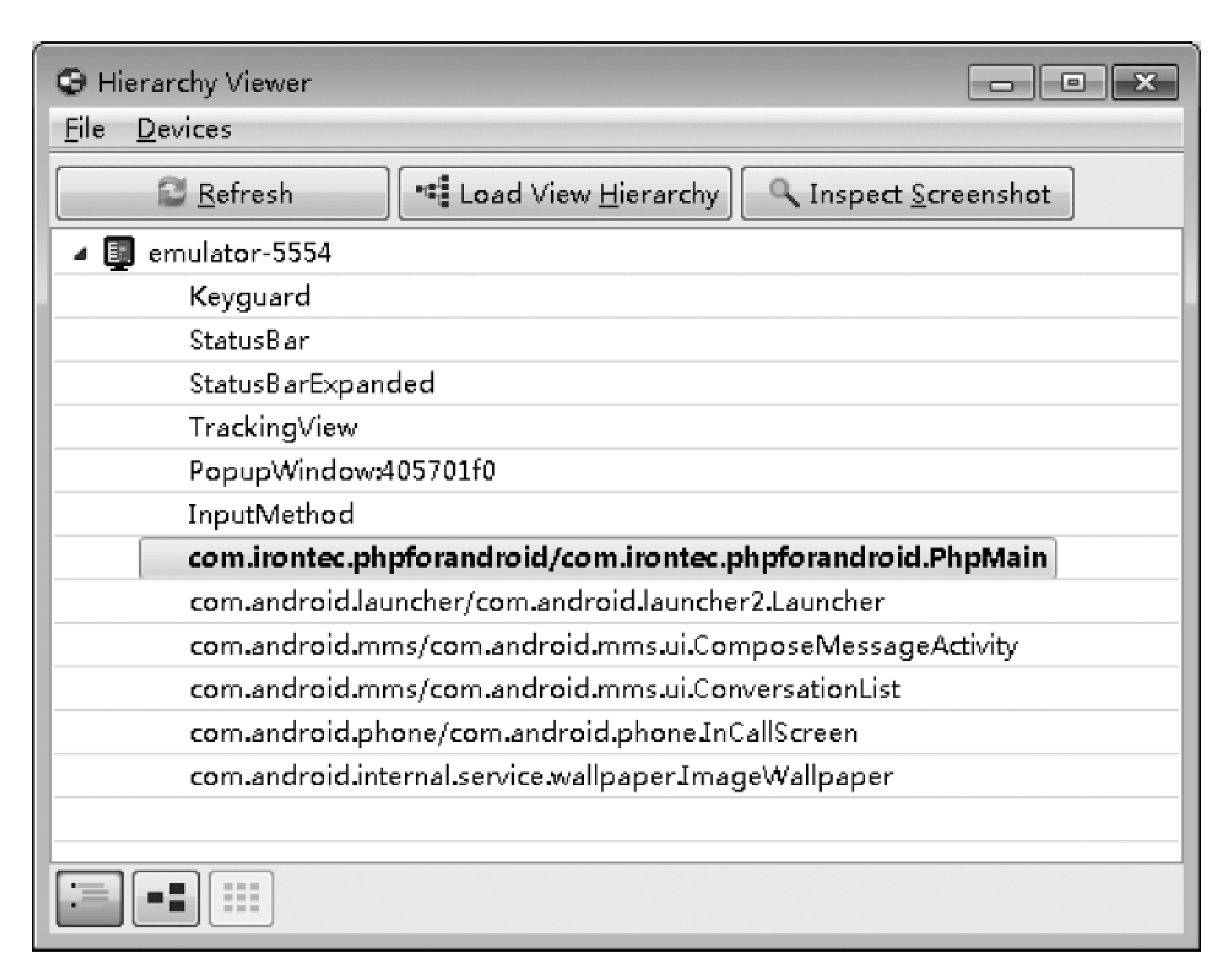


图 16-7-2 Hierarchy Viewer 主界面



图16-7-3 PFA 界面结构和属性信息

显示每个控件的具体属性，是控件的属性面板。而右下角部分的区域，则是当用户单界面中的某个控件时，会在该部分显示区域显示出用户所单击的控件，显示控件在界面中的具体位置时会用红色标出，以方便用户辨识。

接下来查看每个具体控件的情况。当在主窗口中单击每一个控件时，可以看到很多关于这个控件的详细信息，会在该控件的上方弹出一个窗口，其中除显示该控件的实际效果图外，还可以通过 view 的数目显示该控件及其子控件的数目，以及该控件的该节点的测量(measure)、布局(layout)以及画视图(draw)的时间。“1 view”表明这个控件不包

含其他子控件，只有 1 个控件，就是它本身，大于 1 则表明该控件还包含了其他控件。而下方的带颜色的 3 个圆圈指示灯分别说明了在测量(measure)、布局(layout)以及画视图(draw)3 个阶段这个控件所占用的时间百分比，如果是绿色的，表示该控件在该阶段比其他 50%的控件速度要快，为黄色的表示比其他 50%的控件速度要慢，为红色的则表示该控件在该阶段的处理速度是最慢的。单击主窗口布局中显示的第一个按钮，该控件在测量(measure)、布局(layout)以及画视图(draw)3 个阶段所耗费时间为 0.211ms、0.032ms 和 2.150ms，该控件运行速度偏快，该 Button 控件没有包含其他控件，运行结果如图 16-7-4 所示。

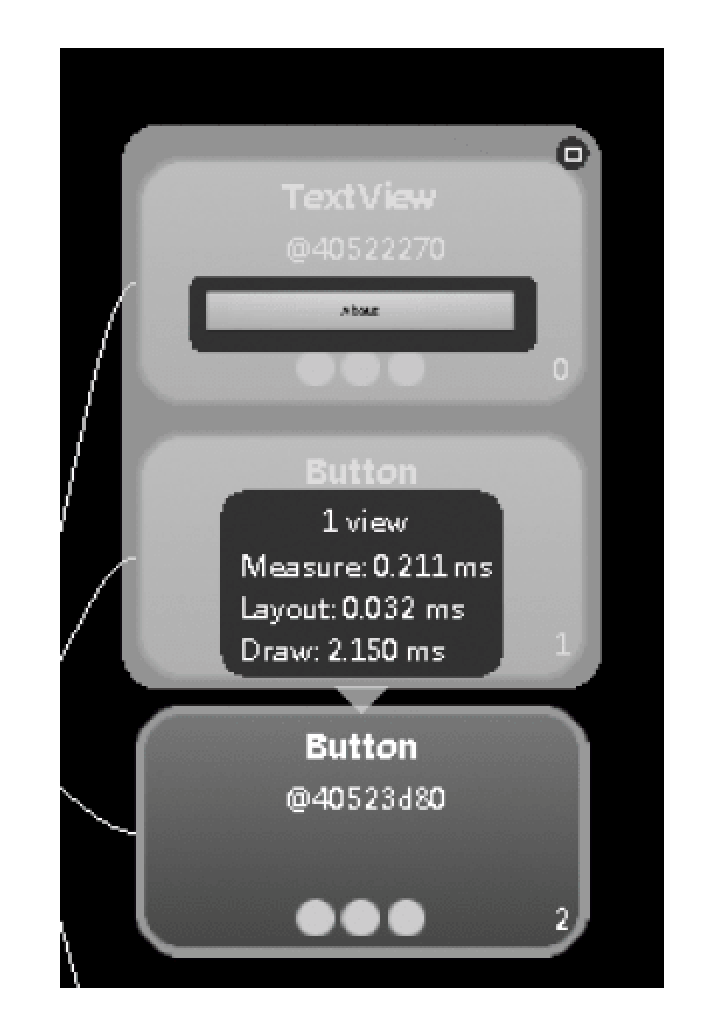


图 16-7-4 PFA 程序 Button 控件信息

16.8 zipalign 工具

在 Android 中，每个应用程序中储存的数据文件都会被多个进程访问：安装程序会读取应用程序的 manifest 文件来处理与之相关的权限问题；Home 应用程序会读取资源文件来获取应用程序的名称和图标；系统服务会因为很多原因读取资源(例如，显示应用程序的 Notification)；此外，就是应用程序自身用到资源文件。

对于未整理的应用程序包，资源读取缓慢，程序内存(RAM)的使用会处在一个较高的范围。资源读取性能和内存使用容量也取决于当前有多少未整理的应用程序。例如，如果系统的应用程序较少，而且有一个未整理的主界面程序，那么在系统启动时能观察到更慢的应用程序。更加糟糕的是，如系统有许多未整理的应用程序，会导致系统反复地启动和结束进程，系统运行将会滞后，电池的使用时间会大幅度降低。对程序包进行优化是解决这个问题的有效方法。

Android SDK 中包含一个 zipalign 工具，它能够对打包的应用程序进行优化，让应用程序和整个系统运行得更快。apk 包的本质是一个 zip 压缩文档，优化的目的是使包内未压缩的数据能够有序地排列，从而减少应用程序运行时的内存消耗。

使用 zipalign 对应用程序中的资源做对齐操作很简单，可以在 DOS 窗口下输入如下命令对程序包进行优化：

```
zipalign -v 4 source.apk destination.apk
```

其中-v 代表详细输出，4 代表对齐为 4B，source.apk 和 destination.apk 分别代表源 apk 和目标 apk 文件。注意：优化必须在签名 apk 文件后进行。

以下的命令用于检查程序包是否进行了对齐：

```
zipalign -c -v 4 application.apk
```

这里-c代表检查对齐。

16.9 monkey工具

16.9.1 monkey简述

monkey是Android中的一个命令行工具,可以运行在模拟器或实际设备中。它向系统发送伪随机的用户事件流(如按键输入、触摸屏输入和手势输入等),实现对正在开发的应用程序进行压力测试。可以使用monkey工具对程序的稳定性和健壮性进行测试。

monkey测试的对象仅为应用程序包,测试使用的事件流数据流是随机的,不能进行自定义,测试的事件数量、类型和频率等是可以设置的。

16.9.2 monkey的基本用法

要运行monkey命令,可以通过在每条monkey命令前加上adb shell来达到目的,也可以进入shell后直接输入monkey命令。基本语法如下:

```
adb shell monkey [options]
```

如果不指定options,monkey将以无反馈模式启动,并把事件任意发送到安装在目标环境中的全部软件包。

下面给出使用monkey工具测试Android模拟器中自带的计算器程序的实例,它会启动指定的应用程序Calculator,并向其发送500个伪随机事件(包括按键、touch事件和系统事件等),在模拟器中将看到程序自动启动并随机输入内容,测试命令如下:

```
adb shell monkey -p com.android.calculator2 -v 500
```

其中参数-p用来指定程序包,com.android.calculator2表示程序Calculator的包名,参数-v表示事件数量。

16.9.3 monkey的选项

monkey提供了4类选项:常规、事件、约束和调试。下面是常规类选项说明。

(1) -help。列出简单的用法。

(2) -v。命令行的每一个-v将增加反馈信息的级别。Level 0(默认值)只提供较少信息,Level 1提供较为详细的测试信息,Level 2提供更加详细的设置信息。

事件类选项说明。

(3) -s。伪随机数生成器的seed值。如果用相同的seed值再次运行monkey,那么它将生成相同的事件序列。

(4) -throttle。在事件之间插入固定延迟。通过这个选项可以减缓monkey的执行速度。如果不指定该选项,monkey将不会被延迟,事件将尽可能快地产生。

(5) -pct-touch。调整触摸事件的百分比(触摸事件是一个 down-up 事件,它发生在屏幕上的某单一位置)。

(6) -pct-motion。调整动作事件的百分比(动作事件由屏幕上某处的一个 down 事件、一系列的伪随机事件和一个 up 事件组成)。

(7) -pct-trackball。调整轨迹事件的百分比(轨迹事件由一个或几个随机的移动组成,有时还伴随有单击)。

(8) -pct-nav。调整"基本"导航事件的百分比(导航事件由来自方向输入设备的 up、down、left、right 组成)。

(9) -pct-majornav。调整"主要"导航事件的百分比(这些导航事件通常引发图形界面中的动作,如 5-way 键盘的中间按键、回退按键和菜单按键)。

(10) -pct-syskeys。调整"系统"按键事件的百分比(这些按键通常被保留由系统使用,如 Home、Back、Start Call、End Call 及音量控制键)。

(11) -pct-appswitch。调整启动 Activity 的百分比。

(12) -pct-anyevent。调整其他类型事件的百分比。包括按键和其他不常用的设备按钮等事件。

-p 属约束限制类。如果用此参数指定了一个或几个包,那么 monkey 将只允许系统启动这些包里的 Activity。要指定多个包,需要使用多个-p 选项,每个-p 选项只能用于一个包。

16.9.4 monkey 测试的停止条件

monkey 执行过程中在下列 3 种情况下会自动停止:

(1) 如果限定了 monkey 运行在一个或几个特定的包上,那么它会监测试图转到其他包的操作,并对其加以阻止。

(2) 如果应用程序崩溃或接收到任何失控异常,那么 monkey 将停止并报错。

(3) 如果产生了应用程序不响应(application not responding)的错误,那么 monkey 将会停止并报错。

通过多次不同设定下的 monkey 测试,程序才算是一个具有足够稳定性的程序。

第 17 章



界面、电池和系统管理 API

17.1 PHP API for Android

除了可以用 Java 来编写 Android 应用外，还可以使用 PHP 编写 Android 应用，这都要归功于一个开源项目 PFA(PHP For Android)。它实际上是利用了另外一个开源项目 SL4A(Scripting Layer for Android)提供的 Android 接口 API 实现的，任何支持 SL4A 的脚本语言(比如 Javascript、Ruby、Perl、PHP 和 Python)都能够通过 Android 接口 API 直接跟操作系统打交道编写应用。事实上，SL4A 只提供了 Android 接口 API 的子集，并没有提供全部 Android 接口 API，尽管是这样，但其功能已经足够强大，可以通过这些接口实现电话、短信、界面、联系人、电池、蓝牙、WiFi、多媒体、摄像、事件、多媒体播放器、传感器和程序首选项等功能和应用。可以通过设备或模拟器上的 PHP 脚本编辑器帮助菜单查看所有的 API 文档。Android 接口 API 被调用后，其运行结果都会返回一个对象，该对象包含 3 个属性：id、result 和 error。id 是一个和 API 调用相关的递增的数字，用于标识对象。result 是 API 调用的返回结果，如果没有返回结果，其值为 null。error 表示 API 调用错误描述，当没有错误时其值为 null。

下面是 PHP 调用 Android API 的范例，例中先包含使用 Android 接口 API 必须引用的文件 Android.php，然后创建 Android 对象，再通过该对象中的方法函数 makeToast 在屏幕上显示字符串“Hello World”，最后调用 print_r 显示函数 makeToast 的返回内容。

【例 17-1-1】 (代码位置：\17\test.php)

```
<?php
  require_once("Android.php");              //包含开发 Android 应用必需的文件 Android.php
  $droid=new Android();                     //创建 Android 对象
  $obj=$droid->makeToast("Hello World");              //调用 Android 对象中的方法 makeToast
  print_r($obj);                                      //显示方法 makeToast 的返回结果
?>
```

先把上面的代码保存到计算机中，文件名为 test.php，然后通过命令 adb push test.php/sdcard/sl4a/scripts/把文件复制到模拟器中(为更高效地开发，可利用豌豆荚软件通过图形界面管理模拟器 SD 卡中的文件)，再在模拟器中的 SL4A 管理器中单击程序

test. php 的图标运行该程序，运行前选择以终端方式运行(单击左边黑色图标)，最后会看到屏幕中间会显示“Hello World”，同时打印函数 makeToast 的返回内容，其运行过程如图 17-1-1 和图 17-1-2 所示，其中图 17-1-1 是选择以终端方式运行程序 test. php，图 17-1-2 是程序的最终运行结果。

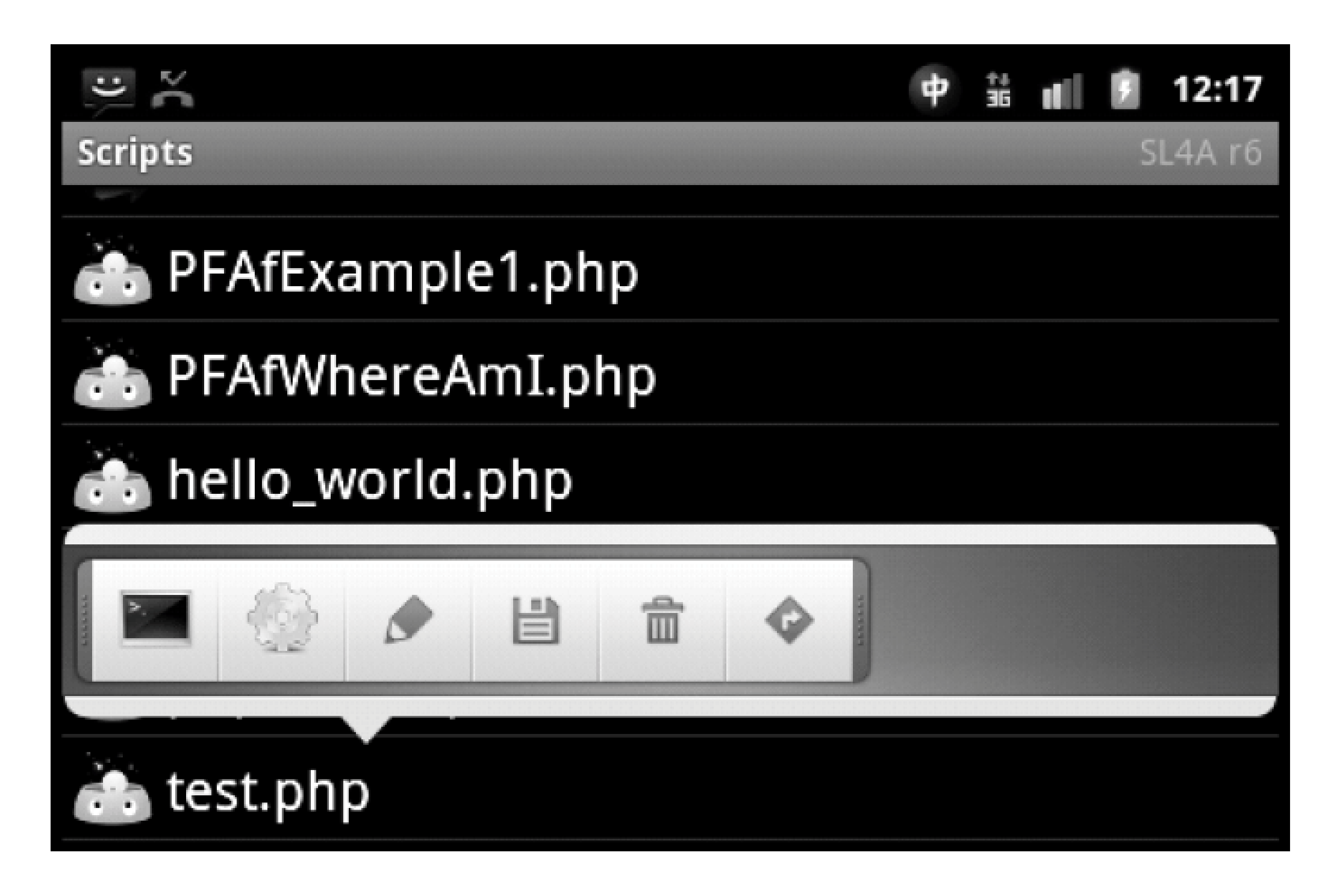


图 17-1-1　选择终端方式

```
FIX ME! implement getprotobyname() bionic/libc/bionic/stubs.c:378
FIX ME! implement getprotobyname() bionic/libc/bionic/stubs.c:378
Array
(
    [id] => 1
    [result] =>
    [error] =>
)
```

Process has exited.
Close terminal?

Hello World

Yes　No

图 17-1-2　test. php 运行结果

17.2　Android 界面 API

17.2.1　警告对话框和对话框按钮

```
dialogCreateAlert(String title[optional],String message[optional])
```

函数 dialogCreateAlert 用于创建对话框，参数 title 表示对话框的标题，参数 message 表示对话框内容，这两个参数是可选的。

```
dialogGetResponse()
```

dialogGetResponse 函数用于等待用户在对话框中输入信息并返回对话框响应。

```
dialogShow()
```

dialogShow 函数用于显示对话框。

```
dialogSetPositiveButtonText(String text)
```

dialogSetPositiveButtonText 函数用于设置对话框确定按钮文字,参数 text 是按钮文字。

```
dialogSetNegativeButtonText(String text)
```

dialogSetNegativeButtonText 函数用于设置对话框否定按钮文字,参数 text 是按钮文字。

```
dialogSetNeutralButtonText(String text)
```

dialogSetNeutralButtonText 函数用于设置对话框中立按钮文字,参数 text 是按钮文字。

下面是对话框的范例,例中先创建对话框和设置 3 个对话框按钮的文字,然后显示对话框窗口,最后等待用户单击按钮并返回用户响应。

【例 17-2-1】 (代码位置:\17\testDlg.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateAlert("是否继续","请选择点击下面按钮");  //设置对话框标题内容
  $droid->dialogSetPositiveButtonText("确定");              //设置"确定"按钮文本
  $droid->dialogSetNeutralButtonText("取消");               //设置"取消"按钮文本
  $droid->dialogSetNegativeButtonText("退出");              //设置"退出"按钮文本
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行结果如图 17-2-1 所示。



图 17-2-1 对话框

17.2.2 请求用户输入信息和密码对话框

```
dialogGetInput(String title,String message,
String defaultText)
```

函数 dialogGetInput 用于请求用户输入信息,参数 title 表示输入框的标题,默认值为 Value;参数 message 表示输入框上的提示信息,默认值为 Please enter value;参数 defaulttext 表示输入框内的提示信息。这 3 个参数都是可选的。

```
dialogGetPassword(String title,String message)
```

函数 dialoggetpassword 用于请求用户输入密码,参数 title 表示密码框的标题,默认值为 Password;参数 message 是显示在密码框上方的提示信息,默认值为 Please enter password 这 2 个参数都是可选的。

下面是登录的范例,例中先请求输入用户名,再请求输入密码。

【例 17-2-2】 （代码位置：\17\testInputPwdDlg. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogGetInput("登录","请输入用户名:","test");
  $droid->dialogGetPassword("登录","请输入密码:");
?>
```

程序运行结果如图 17-2-2 所示，其中图 17-2-2(a)是用户名对话框，图 17-2-2(b)是密码对话框。



(a) 输入用户名



(b) 输入密码

图 17-2-2　登录

17.2.3　列表

```
dialogSetItems(JSONArray items)
```

函数 dialogSetItems 用于设置对话框列表条目，参数 items 表示条目的文本内容，它是一个数组，数组中的每个元素表示一项条目。

下面是列表的范例，例中用数组 items 表示列表内容，共有 4 个列表项，$response["result"]->item 是选项的值，第 1 个列表项对应的值是 0，其他列表项的值依次递增。

【例 17-2-3】 （代码位置：\17\testItemsDlg. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateAlert("列表");
  $items=array();
  $items[0]='Item1';
  $items[1]='Item2';
  $items[2]='Item3';
  $items[3]='Item4';
  $droid->dialogSetItems($items);          //设置一个包含 4 个选项的列表
  $droid->dialogShow();
  $response=$droid->dialogGetResponse();
```

```
  $droid->dialogSetItems(null);
  $item=$response["result"]->item+1;
  $droid->dialogCreateAlert("你选择的列表项","Item$item");
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行后会显示4个菜单项，用户单击Item4选项后，程序会提示用户选择的选项是Item4，其结果如图17-2-3所示，其中图17-2-3(a)是选项列表，列表中有4个选项，图17-2-3(b)显示用户选择的选项。



(a) 选项列表



(b) 显示已选列表项

图 17-2-3 列表操作

17.2.4 单选按钮和复选框

```
dialogSetSingleChoiceItems(JSONArray items,Integer selected)
```

函数dialogSetSingleChoiceItems的作用是设置对话框为单选按钮对话框并选中指定项，参数items为选项列表，它用数组表示，数组中的每个元素表示一个选项；参数selected为指定的选项，默认值为0，它是可选项。单选按钮列表只允许用户从多个选项中选择一个选项。

下面是单选项的范例，例中创建对话框，对话框中有3个单选按钮，默认指定第1个选项为选中项，当用户改变选中项并单击OK按钮后，程序会显示用户选择的选项。

【例 17-2-4】 （代码位置：\17\testSingleChoiceDlg. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateAlert("选择对话框");     //设置对话框标题
  $items=array();
```

```
    $items[]='Choice1';
    $items[]='Choice2';
    $items[]='Choice3';
    $droid->dialogSetSingleChoiceItems($items);      //设置一个包含 3 个选项的单选按钮
    $droid->dialogSetPositiveButtonText("确定");
    $droid->dialogSetNegativeButtonText("退出");
    $droid->dialogShow();                            //显示对话框
    $ret=$droid->dialogGetResponse();                //等待准备响应用户单击选项等事件
    $ret=$droid->dialogGetSelectedItems();           //获取用户的选项
    switch($ret['result'][0])
    {
        case 0: $choice="Choice1";
                break;
        case 1: $choice="Choice2";
                break;
        case 2: $choice="Choice3";
                break;
        default:
                break;
    }
    $droid->dialogCreateAlert("用户选项","你的选项是: ".$choice);
    $droid->dialogSetPositiveButtonText("确定");
    $droid->dialogSetNegativeButtonText("退出");
    $droid->dialogShow();
?>
```

用户选择选项 Choice2 的运行过程如图 17-2-4 和图 17-2-5 所示，其中图 17-2-4 是用户选择了对话框中的选项 Choice2，图 17-2-5 是用户选择选项并单击“确定”按钮后的显示结果。



图 17-2-4　选择第 2 项



图 17-2-5　显示用户选择的单选项

下面介绍复选框对话框，复选框对话框允许用户同时选择多个选项。

```
dialogSetMultiChoiceItems(JSONArray items,JSONArray selected)
```

函数 dialogSetSingleChoiceItems 用于设置对话框为复选框对话框并选中指定项。参数

items 为选项列表;参数 selected 为指定的选项列表,它是可选项,如不设置该参数将不设置指定选项。

下面是多选项的范例,例中创建对话框,对话框中有 3 个复选框,用户选中指定项并单击 OK 按钮后,程序会显示用户选择的选项。

【例 17-2-5】 (代码位置:\17\testMultiChoiceDlg.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateAlert("ChoiceItems");
  $items=array();
  $items[]='Choice1';
  $items[]='Choice2';
  $items[]='Choice3';
  $droid->dialogSetMultiChoiceItems($items);              //设置一个包含 3 个选项的复选框
  $droid->dialogSetPositiveButtonText("OK");
  $droid->dialogSetNegativeButtonText("Quit");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $ret=$droid->dialogGetSelectedItems();
  print_r($ret);
  foreach($ret['result'] as $key=>$value)
  {
      $i=$value+1;
      $choice.="Choice".$i." ";
  }
  $droid->dialogCreateAlert("ChoiceItems","Your Choice: ".$choice);
  $droid->dialogSetPositiveButtonText("OK");
  $droid->dialogSetNegativeButtonText("Quit");
  $droid->dialogShow();
?>
```

程序运行后,用户选择 Choice1 和 Choice2 选项后,程序弹出对话框显示已选择了 Choice1 和 Choice2 选项,运行结果如图 17-2-6 和图 17-2-7 所示。



图 17-2-6 用户选中 2 个选项



图 17-2-7 显示用户选择的选项

注意:无论单选还是复选,要获取用户的选项需要借助函数 dialogGetSelectedItems,而

列表选项需借助 dialogGetResponse 函数。

17.2.5 时间设置对话框

```
dialogCreateTimePicker(Integer hour,Integer minute,Boolean is24hour)
```

函数 dialogCreateTimePicker 用于创建时间设置窗口。参数 hour 表示时钟，默认值为 0；参数 minute 表示分钟，默认值为 0；参数 is24hour 表示是否使用 24 小时制，值 true 表示使用 24 小时制，默认值为 false，即使用 12 小时制。这 3 个参数都是可选的。

下面是时间设置的范例，例中先创建时间设置对话框，在用户完成时间设置后显示用户设置的时间。

【例 17-2-6】 （代码位置：\17\testTimePicker.php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateTimePicker(20,11);
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
  $hour=$ret['result']->hour;
  $minute=$ret['result']->minute;
  $droid->dialogCreateAlert("时间(小时和分钟)","$hour : $minute");
  $droid->dialogSetPositiveButtonText("OK");
  $droid->dialogSetNegativeButtonText("Quit");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
?>
```

程序运行后会显示时间对话框，用户可以单击小时数和分钟数上下的“＋”和“－”按钮调整时间，之后可以单击 Set 按钮设置时间，设置结束将显示用户设置的时间。程序运行结果如图 17-2-8 和图 17-2-9 所示，其中图 17-2-8 是设置时间，图 17-2-9 是显示时间。



图 17-2-8　时间设置窗口



图 17-2-9　显示用户设置的时间

17.2.6 日期设置对话框

```
dialogCreateDatePicker(Integer year,Integer month,Integer day)
```

函数 dialogCreateDatePicker 用于创建日期设置窗口。参数 year 表示年，默认值为 1970；参数 month 表示月，默认值为 1；参数 day 表示日，默认值为 1。这 3 个参数可选。

下面是日期设置的范例，例中先创建日期设置对话框，然后用户完成日期设置，最后显示用户设置的日期。

【例 17-2-7】 （代码位置：\17\testDatePicker. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateDatePicker(2000,1,1);
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
  $year=$ret['result']->year;
  $month=$ret['result']->month;
  $day=$ret['result']->day;
  $droid->dialogCreateAlert("Date","$year-$month-$day");
  $droid->dialogSetPositiveButtonText("OK");
  $droid->dialogSetNegativeButtonText("Quit");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
?>
```

程序运行后会显示日期对话框，用户可以通过日期上下的“＋”和“－”按钮调整日期，之后可以单击 Set 按钮设置日期，设置结束后系统将显示用户之前显示的日期。程序运行结果如图 17-2-10 和图 17-2-11 所示，其中图 17-2-10 是设置日期，图 17-2-11 用于显示用户设置的日期。



图 17-2-10 设置日期



图 17-2-11 显示日期

17.2.7 水平进度条

应用程序经常会执行一些需要消耗一定时间的任务，用户希望能知道任务的进度，Android 提供了带有进度指示器的水平进度条用来显示任务的进度位置。例如播放音乐或观看视频时，可以用水平进度条显示当前的播放位置。再比如，水平进度条可以显示文件下载进度情况。PHP For Android 提供 dialogCreateHorizontalProgress 和 dialogSetCurrentProgress 函数用来创建水平进度条对话框和设置水平进度条指示器位置。

```
dialogCreateHorizontalProgress(String title,String message,Integer maximum)
```

函数 dialogCreateHorizontalProgress 用于创建水平进度对话框。参数 title 表示标题；参数 message 表示对话框中显示的内容；参数 maximum 表示进度条的最大值，默认值是 100。这 3 个参数都是可选的。

```
dialogSetCurrentProgress(Integer current)
```

函数 dialogSetCurrentProgress 用于设置进条度的当前值。参数 current 表示当前值，其值应大于或等于零，且小于或等于最大值 maximum。

```
dialogDismiss()
```

函数 dialogDismiss 用于关闭对话框。

下面是进度条的范例。

【例 17-2-8】 （代码位置：\17\testHProgressDlg. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateHorizontalProgress("ProgressBarTitle","Please Wait...");
  $droid->dialogShow();
  for($items=0;$items<=100;$items++)
    $droid->dialogSetCurrentProgress($items);
  $droid->dialogDismiss();
?>
```

程序运行后会显示进度条对话框，对话框中的进度条会不断前进，当进度值达到 100 后，进度条对话框将消失，运行结果如图 17-2-12 所示。

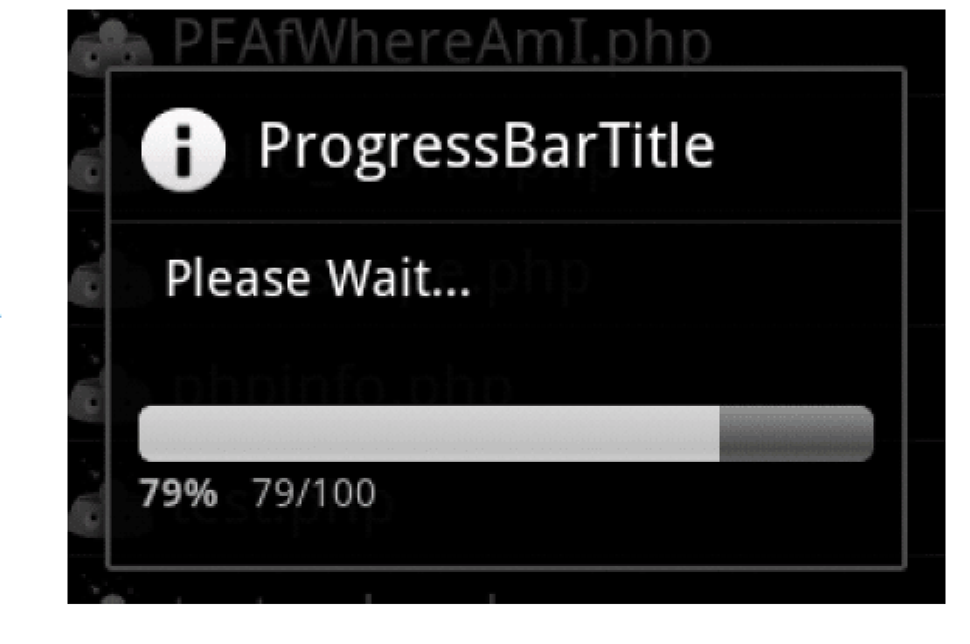


图 17-2-12 进度条对话框

17.2.8 搜索进度条

在观看视频时经常要拖动指示器改变播放内容或调节音量，那该怎么实现呢？PFA 提供了搜索进度条让用户拖动指示器来实现这样的

功能。从外观上看，搜索进度条和水平进度条相同，但不同的是，搜索进度条可以让用户以手工拖动方式改变进度条的进度位置，而水平进度条不允许用户手工改变进度。

```
dialogCreateSeekBar(Integer starting,Integer maximum,String title,String message)
```

函数 dialogCreateSeekBar 用来创建搜索进度条。参数 starting 表示当前值，默认值为 50；参数 maximum 表示进度条最大值，默认值为 100；参数 title 表示对话框标题；参数 message 表示对话框显示信息。

下面是搜索进度条的范例，例中创建了一个搜索进度条，进度条中有一个水平滑动条，用户可以单击滑动条自行设置滑动条的滑动位置，单击"是"按钮后将显示滑动位置值。

【例 17-2-9】 （代码位置：\17\testSeekBar.php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateSeekBar(50,100,"搜索条对话框标题","搜索条对话框显示信息");
  $droid->dialogSetPositiveButtonText("是");
  $droid->dialogSetNegativeButtonText("否");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
  $msg=$ret['result']->progress;
  $droid->dialogCreateAlert("搜索条值","$msg");
  $droid->dialogSetPositiveButtonText("是");
  $droid->dialogSetNegativeButtonText("否");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
?>
```

程序运行结果如图 17-2-13 和图 17-2-14 所示，其中图 17-2-13 是用户设置的进度值，图 17-2-14 显示了该值。



图 17-2-13　搜索进度条



图 17-2-14　显示搜索进度条进度值

17.2.9 等待完成进度条

当应用程序去连接网络时,用户需要等待,但等待时间是不确定的。类似这种应用,可以用一个带有旋转圆圈的进度条提示用户系统正在处理之中,避免用户感觉应用已经“死机”,旋转的圆圈表示有任务在进行中但完成的时间不确定。

```
dialogCreateSpinnerProgress(String title,String message,Integer maximum)
```

dialogCreateSpinnerProgress 函数用来创建等待完成进度条。参数 title 表示对话框标题;参数 message 表示进度条的显示内容;参数 maximum 表示进度值,默认值为 100。这 3 个参数都是可选的。

下面是等待完成进度条的范例,例中创建了一个网络连接等待完成进度条,用户等待 5s 后该进度条自动消失。

【例 17-2-10】 (代码位置:\17\testSpinnerProgress.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateSpinnerProgress("网络连接","正在连接中,请耐心等待...");
  $droid->dialogShow();
  for($i=0;$i<5;$i++)
    sleep(1);          //可以在这写网络连接代码,这里使用睡眠函数 sleep 模拟处理
  $droid->dialogDismiss();
?>
```

程序运行后,会看到一个带有旋转的圆圈的对话框,运行结果如图 17-2-15 所示。



图 17-2-15 等待完成对话框

17.3 首选项 API

对于需要跨应用程序执行期间或生命期而维护重要信息的应用程序来说,能够在移动设备上本地存储数据是非常关键的功能。开发人员经常需要存储诸如用户首选项或应用程序配置之类的信息。在用户看来,首选项就是一些文本框、复选框和下拉菜单等的集合体。而从数据的角度看来,首选项是一组由键值对构成的数据集,其中的值都是基本数据类型,如整型、布尔型和字符串等。

首选项这个词是 preference 翻译过来的。首选项类似于 Windows 中的 ini 文件和

Web 中的 Cookie,它的本质是基于 XML 文件存储 key-value 键值对数据。通常用它来保存一些配置文件、用户名和密码等简单的配置信息,其默认存储位置在模拟器或手机中的/data/data/com.googlecode.android_scripting/shared_prefs 目录下。首选项只能在同一个包中使用,不同包之间是不能共享首选项存储的数据的。同时,首选项存储的文件是不能存储到 SD 上的,这是首选项的一个局限。首选项存储方式的优点是存储数据简单、方便和快捷,适用于存储少量的简单配置信息,不适用于存储数据量较大的信息。PFA 提供了 preferences API 用于方便地进行数据存储,它是一种轻量级的数据存储方式。preferences API 包含 prefPutValue、prefGetValue 和 prefGetAll 等函数,下面介绍这 3 个函数。

```
prefPutValue(String key,Object value,String filename)
```

prefPutValue 用于存储键值对。参数 key 表示键名;参数 value 表示键值,它可以是整数、浮点数和字符串等数据类型;参数 filename 表示存储键值对的文件,该参数是可选的。如果没有指定 filename 则使用默认的共享首选项文件名。

```
prefGetValue(String key,String filename)
```

prefGetValue 用于从指定文件中读取指定键的值。参数 key 表示键名;参数 filename 表示存储键值对的文件,该参数是可选的。如果没有指定 filename 则使用默认的共享首选项文件名,如果找到指定的键名则返回对应的键值,否则为 null。

```
prefGetAll(String filename)
```

prefGetAll 用于从指定文件中读取所有的键值对。参数 filename 表示存储键值对的文件,该参数是可选的。如果没有指定 filename 则使用默认的共享首选项文件名。

下面是首选项的范例,例中先把键值对写入到 MyPref.xml 文件中,然后再从该文件中读出键名为 mykey 的键值并显示出来。

【例 17-3-1】 (代码位置:\17\testPref1.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $key="mykey";
  $value="hello";
  $filename="MyPref";
  $droid->prefPutValue($key,$value,$filename);
  $ret=$droid->prefGetValue($key,$filename);
  $droid->dialogCreateAlert("程序首选项演示(键:值)",$key.":".$ret['result']);
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行结果如图 17-3-1 所示。



图 17-3-1 首选项演示

同时，程序会在目录/data/data/com. googlecode. android_scripting/shared_prefs 下生成文件 MyPref. xml，其内容如下：

```
<?xml version='1.0' encoding='utf-8'
standalone='yes' ?>
<map>
<string name="mykey">hello</string>
</map>
```

下面是用户设置音量的范例，例中先从文件 MyPref. xml 读出 voice 键值，该键值为音量值，如果没能读出该键值则音量值使用默认值 50，然后使用搜索进度条显示音量，用户可以改变该值，最后使用首选项 API 保存用户设置的音量值。

【例 17-3-2】 (代码位置：\17\testPref2. php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $key="voice";
  $filename="MyPref";
  $ret=$droid->prefGetValue($key,$filename);
  $progress=$ret['result'];
  if(null==$progress)
    $progress=50;
  $droid->dialogCreateSeekBar($progress,100,"模拟音量设置","音量值:$progress");
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("取消");
  $droid->dialogShow();
  $ret=$droid->dialogGetResponse();
  $droid->dialogDismiss();
  $progress=$ret['result']->progress;
  $droid->prefPutValue($key,$progress,$filename);
?>
```

程序运行后，用户可打开文件 MyPref. xml 查看属性 voice 和值，检验其值是否是用户设置的值。

下面是 prefGetAll 的示例，例中会读出所有的键值对并打印出来。

【例 17-3-3】 (代码位置：\17\testPref3. php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $filename="MyPref";
```

```
  $ret=$droid->prefGetAll($filename);
  print_r($ret);
  echo $ret['result']->voice;
  echo $ret['result']->mykey;
?>
```

程序以终端方式运行,其运行结果如图 17-3-2 所示。

```
Array
(
    [id] => 1
    [result] => stdClass Object
        (
            [voice] => 28
            [mykey] => hello
        )

    [error] =>
)
28hello
```

图 17-3-2 prefGetAll 示例结果

17.4 电池管理 API

移动设备通过电池维护其工作,虽然许多应用程序并不需要知晓电池的状态,但是某些应用程序可能需要根据电池的电量、充电状态或电源管理设置来改变自己的行为。GPS是相当耗电的,类似这种耗电的应用通常要监视电池当前的电量,在电量达到一个下限的时候,及时提醒用户,使用户能根据情况决定是否关闭GPS,这样可节省电量以备电话等急用。通过电池的电量级别还可以监测应用程序的执行效率,开发人员可以借此来优化应用。再比如,程序监视到电池电量较低时可以降低其扫描工作频率,而当监视到使用外部电源补充电量后又可以提高其工作频率。这样做会受到用户的欢迎。PFA提供了电池监视API,这些API用来监视电池的电压、健康状态、电量级别、充电状态和温度等信息,接下来介绍电池监视API。

```
batteryCheckPresent()
```

函数 batteryCheckPresent 用于查看电池的最近电量信息,它将返回当前的电池电量信息。

```
batteryGetHealth()
```

函数 batteryGetHealth 用于获取最近接收到的电池健康状态,返回值和状态如表 17-4-1 所示。

表 17-4-1 电池健康状态表

返回值	1	2	3	4	5	6
状态	未知	良好	过热	不可用	电池过饱	查询失败

```
batteryGetLevel()
```

函数 batteryGetLevel 用于查询电池的电量,返回值格式为百分比。

```
batteryGetPlugType()
```

函数 batteryGetPlugType 用于查询充电电源信息,返回值和状态如表 17-4-2 所示。

表 17-4-2 电池电源状态表

返回值	−1	0	1	2
状态	未知	未插入电源	交流电源充电	Usb 充电

```
batteryGetStatus()
```

函数 batteryGetStatus 用于查询电池充电状态，返回值和状态如表 17-4-3 所示。

表 17-4-3　电池充电状态表

返回值	1	2	3	4	5
状态	未知	正在充电	放电中	未充电	已充满

```
batteryGetTechnology()
```

函数 batteryGetTechnology 用于查询电池技术信息。

```
batteryGetTemperature()
```

函数 batteryGetTemperature 用于查询电池温度，返回值为当前电池的温度。

```
batteryGetVoltage()
```

函数 batteryGetVoltage 用于查询电池的电压，返回值为当前电池的电压。

```
batteryStartMonitoring()
```

函数 batteryStartMonitoring 用于开始监视电池状态，在查询电池的各种状态之前应该先调用该函数，否则查询无效。

```
batteryStopMonitoring()
```

函数 batteryStopMonitoring 用于停止监视电池状态，在结束查询电池状态之后应该调用该函数停止监视。

```
readBatteryData()
```

函数 readBatteryData 用于读取电池记录数据，这些数据包含电池充电、电池电源、电池健康、电池温度和电池技术等数据。

下面是检测电池健康状态的范例，例中调用 batteryGetHealth 函数获取电池健康状态值并用对话框显示该值。

【例 17-4-1】（代码位置：\17\testBattery1.php）

```
<?php
  require_once("Android.php");
$droid=new Android();
  $droid->batteryStartMonitoring();
  $ret=$droid->batteryGetHealth();
  $droid->batteryStopMonitoring();
  $droid->dialogCreateAlert("电池健康状态检测","返回值: ".$ret['result']);
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
```

```
  $droid->dialogGetResponse();
?>
```

程序运行结果如图 17-4-1 所示。



图 17-4-1 电池健康状态值

下面是检测电池各种状态的范例，例中调用 readBatteryData 函数获取电池各种状态值并用列表显示出来。

【例 17-4-2】（代码位置：\17\testBattery2.php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->batteryStartMonitoring();
  $ret=$droid->readBatteryData();
  $droid->batteryStopMonitoring();
  $droid->dialogCreateAlert("电池状态");
  $items=array();
  $items[0]="level:".$ret['result']->level;
  $items[1]="technology:".$ret['result']->technology;
  $items[2]="status:".$ret['result']->status;
  $items[3]="voltage:".$ret['result']->voltage;
  $items[4]="plugged:".$ret['result']->plugged;
  $items[5]="batter_present:".$ret['result']->batter_present;
  $items[6]="health:".$ret['result']->health;
  $items[7]="temperature:".$ret['result']->temperature;
  $droid->dialogSetItems($items);
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

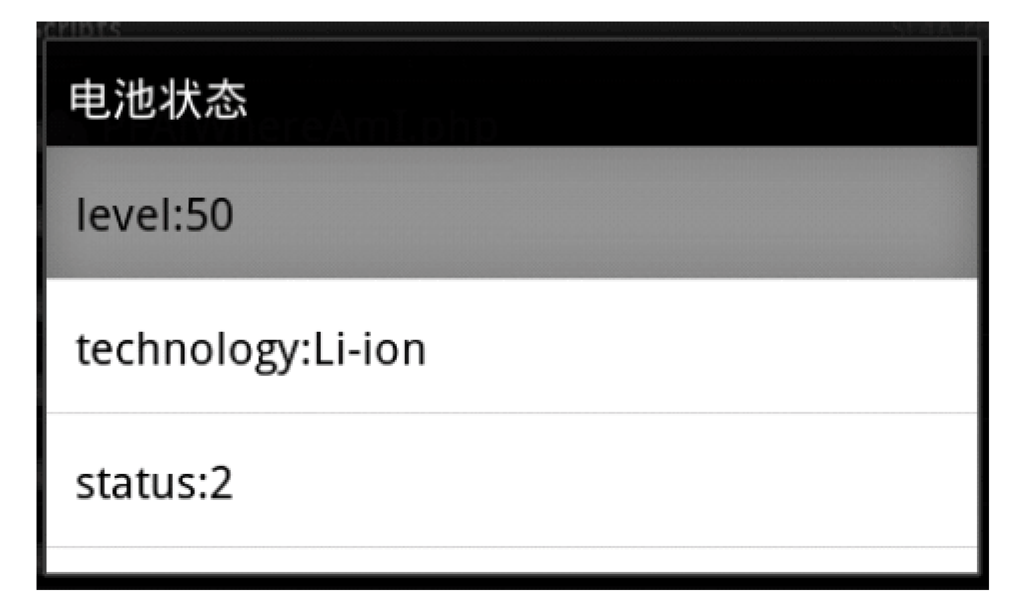


图 17-4-2 电池状态

程序运行后以列表方式显示所有电池状态，每个列表项表示一项电池状态，由于屏幕较小，不能同时把所有电池状态显示在屏幕上，可通过鼠标滑动轮或键盘向下方向键改变列表项焦点来查看全部状态值。运行结果如图 17-4-2 所示。

为了测试程序是否能正常测试电池状态，可以通过控制台命令方式修改模拟器电池状态。在模拟器运行的情况下，可以通过以下 telnet 命令进入修改电池状态控制台界面：

```
telnet localhost 5554
```

5554 表示模拟器端口。进入控制台后，可以输入不同的命令修改电池的健康和电量等信

息。修改电池充电状态命令如下：

```
power status unknown|charging|discharging|not-charging|full
```

例如，命令 power status discharging 表示修改电池充电状态为放电状态。修改电池是否接上交流电电源的命令如下：

```
power ac on|off
```

例如，命令 power ac on 表示电池接上交流电电源。修改电池电压量命令如下：

```
power capacity <percentage>
```

例如，命令 power capacity <80%>表示修改电量为80%。命令 power display 表示查看电池状态，运行命令后结果如下：

```
AC: online
status: Charging
health: Good
present: true
capacity: 80
```

17.5 Android 系统管理 API

17.5.1 环境变量

有时候应用程序需要知道 Android 手机的环境变量值才能完成相应的工作任务，事实上，Android 手机也内置了一些环境变量供程序使用以便统一管理。比如，应用程序要下载保存文件，可以读取文件下载目录环境变量，并把文件保存到下载目录环境变量指定的下载目录中。再比如，当下载或安装一个程序之前，往往会比较关心当前系统内存和 SD 卡上的剩余容量，在做文件复制和移动操作时，检测目标目录的空间大小也是必需的，存储空间大小可以通过环境变量计算出来。因为 Android 是基于 Linux 系统的，所以没有盘符的概念，而且是以存储块来存储数据，存储块大小以字节为单位。PFA 提供了函数 environment 用来获取环境变量。

```
environment()
```

函数执行成功后，会返回缓冲目录、文件下载目录等环境的值。下面是函数 environment 的示例，该范例读取环境变量并以列表方式显示变量名和变量值。例中变量 $ret['result']->appcache 表示获取应用程序的缓冲目录，变量 $ret['result']->download 表示获取文件下载目录，变量 $ret['result']->SDK 表示 SDK 的 API 级别，变量 $ret['result']->sdcard->blocksize 表示获存储块的大小，$ret['result']->sdcard->blockcount 表示存储块数量，$ret['result']->sdcard->availblocks 表示有效存储块数量。

【例 17-5-1】（代码位置：\17\testEnv.php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->environment();
  $droid->dialogCreateAlert("环境变量");
  $items=array();
  $items[0]="缓存目录:".$ret['result']->appcache;
  $items[1]="文件下载目录:".$ret['result']->download;
  $items[2]="SDK API 级别:".$ret['result']->SDK;
  $items[3]="存储块大小:".$ret['result']->sdcard->blocksize;
  $items[4]="存储块数量:".$ret['result']->sdcard->blockcount;
  $items[5]="有效存储块数量:".$ret['result']->sdcard->availblocks;
  $droid->dialogSetItems($items);
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行后，其结果如图 17-5-1 所示，存储块的大小是 512B，共有 258 048 个存储块，220 438 个有效存储块。



图 17-5-1 环境变量

SD 卡存储容量大小可以通过下面的公式计算：

SD 卡空间容量＝存储块数量×存储块大小

SD 卡有效空间容量＝有效存储块数量×存储块大小

17.5.2 剪贴板

在开发一些系统应用的时候，可能需要用到 Android 的剪贴板功能，比如将文本复制到剪贴板或者从剪贴板复制数据等操作。在 Android 平台中每个常规的应用运行在自己的进程空间中，相对于 Win32 而言，Android 上的进程间传递数据方式主要有 IPC 和剪切板，还可以用剪贴板在同一个应用程序的多个活动之间共享数据。剪贴板的使用较简单，可以通过 getClipboard 和 setClipboard 函数获取和设置剪贴板内容。

```
getClipboard()
```

函数 getClipboard 用于从剪贴板中读取文本内容，其返回值是字符串类型。

```
setClipboard(String text)
```

函数 setClipboard 用于将文字内容放到剪贴板中，供自身或其他应用程序使用。参数 text 是放到剪贴板中的文本内容。

下面是剪贴板的范例,例中先把文本放到剪贴板中,然后再从剪贴板中读出文本内容,最后把文本内容显示出来。

【例 17-5-2】 (代码位置:\17\testClipboard.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->setClipboard("这里是剪贴板的测试内容");
  $ret=$droid->getClipboard();
  $droid->dialogCreateAlert("剪贴板内容",$ret['result']);
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行结果如图 17-5-2 所示。

剪贴板内容
这里是剪贴板的测试内容
确定 退出

图 17-5-2 剪贴板

17.5.3 类常量

```
getConstants(String classname)
```

函数 getConstants 用来获取类中的 static final 类型常量。参数 classname 表示类名,这里的类是指 Java 类,而不是 PHP 类。

下面是 getConstants 的范例,例中获取了类 android.graphics.Color 中的 BLUE、WHITE 和 BLACK 等常量值并显示出来。

【例 17-5-3】 (代码位置:\17\testConstants.php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->getConstants("android.graphics.Color");
  $droid->dialogCreateAlert("android.graphics.Color 类常量");
  $items=array();
  $items[0]="蓝色值:".$ret['result']->BLUE;
  $items[1]="白色值:".$ret['result']->WHITE;
  $items[2]="透明值:".$ret['result']->TRANSPARENT;
  $items[3]="黄色值:".$ret['result']->YELLOW;
  $items[4]="黑色值:".$ret['result']->BLACK;
  $items[5]="绿色值:".$ret['result']->GREEN;
  $droid->dialogSetItems($items);
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

程序运行结果如图 17-5-3 所示。



图 17-5-3 获取类常量

17.5.4 获取包版本和代号

可以使用函数 getPackageVersion 和 getPackageVersionCode 分别获取包的版本和代号。

17.5.5 通知

Android 设备上用于显示通知和指示器的标准区域位于屏幕顶端的状态栏，状态栏通常用来显示日期、时间和短消息通知，同时，开发人员可以在状态栏上告知用户重要的事件发生了，这能增强应用程序的功能。例如，当一个下载任务完成时可以向用户发送文本通知；再比如有短消息到来时，可以在状态栏上显示通知。可以使用 notify 函数发送文本通知并显示在状态栏上。

```
notify(String title,String message)
```

参数 title 表示消息的标题；参数 message 表示消息的内容。

下面是通知的范例。

【例 17-5-4】 (代码位置：\17\testNotify. php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->notify("消息标题","消息内容");
?>
```



图 17-5-4 通知消息

程序运行后将看到状态栏上出现了通知图标，结果如图 17-5-4 所示。

17.5.6 意图

Intent 也叫意图，Android 中提供了 Intent 机制协助应用间的交互与通信，这种 Intent 机制实质是一种通用的消息机制，消息机制能较好地实现松耦合，体现松耦合性不仅在于消息的异步性，而且提供了 Intent 对象来描述消息从而具有统一性和扩展性。Intent 对象负责描述消息，Android 则根据此 Intent 对象的描述，负责找到对应的组件，将 Intent 对象中的数据传递给调用的组件并完成组件的调用。Intent 不仅可以用于应用程序之间，也可以用于应用程序内部的活动之间的交互。总之，Intent

起中介的作用，专门负责组件间的协调和传递消息，实现调用者与被调用者间的解耦合。

Intent 对象的核心由动作和数据两部分组成。动作即为 Intent 要采取的行动，动作用一个字符串表示，Android 自身提供了很多标准动作用来浏览网页和拨打电话等，也可以自定义 Intent 的动作。数据是动作要处理的数据，其数据格式使用 URI 方式表示，URI 标明了某个实体的位置和名称。例如，如果动作为 android. intent. action. CALL，该动作表示打电话，那么数据将为“tel：电话号码的 URI”；如果动作为 android. intent. action. VIEW，该动作表示查看，则数据将为“http：网络地址的 URI”。一个 Intent 对象的含义可以简单地理解为对“谁”（即 URI 所描述的资源）要“做什么”（即需要执行的动作）。动作是抽象的，它需要组件来实现，同一个动作可以由多个类似的组件来完成，但用户需要从中选择某个组件来完成。例如，当出现接电话动作时，如果手机上装了多个电话程序，那么需要由用户来选择某个电话程序来完成接听动作。下面介绍相关意图函数。

```
makeIntent(String action,String uri,String type,JSONObject extras,JSONArray categories,
String packagename,String classname,Integer flags)
```

函数 makeIntent 用来创建 Intent 对象。参数 action 表示动作，一个 Intent 对象只能声明一个动作；参数 uri 表示数据；参数 type 表示数据的数据类型，如果没有提供 type，系统将自行判断其数据类型；参数 extras 表示其他附加信息集合，它用数组来存储信息，一个数组元素表示一条信息，信息可由基本数据类型或对象类型的数据表示；参数 categories 表示类型，它用于说明处理该 Intent 的组件的种类，一个种类用一个字符串表明，一个 Intent 对象可以有任意个种类；参数 packagename 表示包名，如果书写包名则必须填写类名；参数 classname 表示类名；参数 flags 表明标志，各种标志位用来说明 Android 系统如何启动运行组件，例如，可以设置该标志使执行意图的组件和发送意图的组件运行在两个不同的进程。除参数 action 是必填的以外，其他参数均是可选的。

```
startActivityIntent(Intent intent,Boolean wait)
```

函数 startActivityIntent 用指定的 Intent 对象启动活动。参数 intent 表示意图对象，该对象由函数 makeIntent 创建；参数 wait 是布尔值，该参数是可选的，表示是否以堵塞方式运行，如果值为 true 表示程序将堵塞直到用户结束运行的活动，否则不堵塞。

下面是 Intent 的范例，例中先创建了一个动作为 android. intent. action. VIEW 和数据为 http://www. baidu. com 的 Intent 对象，然后通过函数 startActivityIntent 把该 Intent 对象提交给 Android 系统处理，Android 系统会自动寻找匹配的组件来执行该意图。

【例 17-5-5】（代码位置：\17\testMakeIntent1. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->makeIntent("android.intent.action.VIEW","http://www.baidu.com");
  $droid->startActivityIntent($ret['result']);
?>
```

运行该程序后，系统会自动打开浏览器并访问百度网站。

在手机聊天室中传输图片时可以用非堵塞方式启动活动，实现一边下载文件一边进行其他的事情。下面是启动活动的范例，范例用 Intent 打开百度网站，同时，还继续显示进度条的进度。

【例 17-5-6】 （代码位置：\17\testMakeIntent2. php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->makeIntent("android.intent.action.VIEW","http://www.baidu.com");
  $droid->startActivityIntent($ret['result'],false);
  $droid->dialogCreateHorizontalProgress("ProgressBarTitle","Please Wait...");
  $droid->dialogShow();
  for($items=0;$items<=100;$items++)
    $droid->dialogSetCurrentProgress($items);
  $droid->dialogDismiss();
?>
```

程序运行结果如图 17-5-5 所示。



图 17-5-5　非堵塞方式启动活动

Android 系统提供了标准的动作，下面列出部分常用的标准动作，具体如表 17-5-1 所示。

表 17-5-1　标准动作

动　　作	动 作 说 明
android. intent. action. VIEW	显示指定数据
android. intent. action. DATA	用于指定一些数据被附加的地方，例如，图片数据应该附属于联系人

续表

动　　作	动 作 说 明

android. intent. action. EDIT	编辑指定数据
android. intent. action. PICK	从列表中选择一个子项目,并返回所选中的项目
android. intent. action. CHOOSER	显示一个 activity 选择器,允许用户在进程之前选择他们想要的活动
android. intent. action. GET_CONTENT	允许用户选择特殊种类的数据(如一张相片或一段录音)并返回
android. intent. action. DIAL	拨打一个指定的号码,显示一个带有号码的用户界面,允许用户启动呼叫
android. intent. action. CALL	根据指定的数据执行一次呼叫
android. intent. action. SEND	传递数据,被传送的数据没有指定,接收的 action 请求用户发数据
android. intent. action. SENDTO	发送一个信息到指定的某人
android. intent. action. ANSWER	处理一个打进来的电话呼叫
android. intent. action. INSERT	插入一条空项目到已给的容器中
android. intent. action. DELETE	从容器中删除已给的数据

下面是拨打电话意图的范例。

【例 17-5-7】 (代码位置:\17\testMakeIntent3. php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->makeIntent("android.intent.action.CALL","tel:5556");
  $droid->startActivityIntent($ret['result']);
?>
```

程序运行后自动实现输入手机号码和拨号呼叫,结果如图 17-5-6 所示。

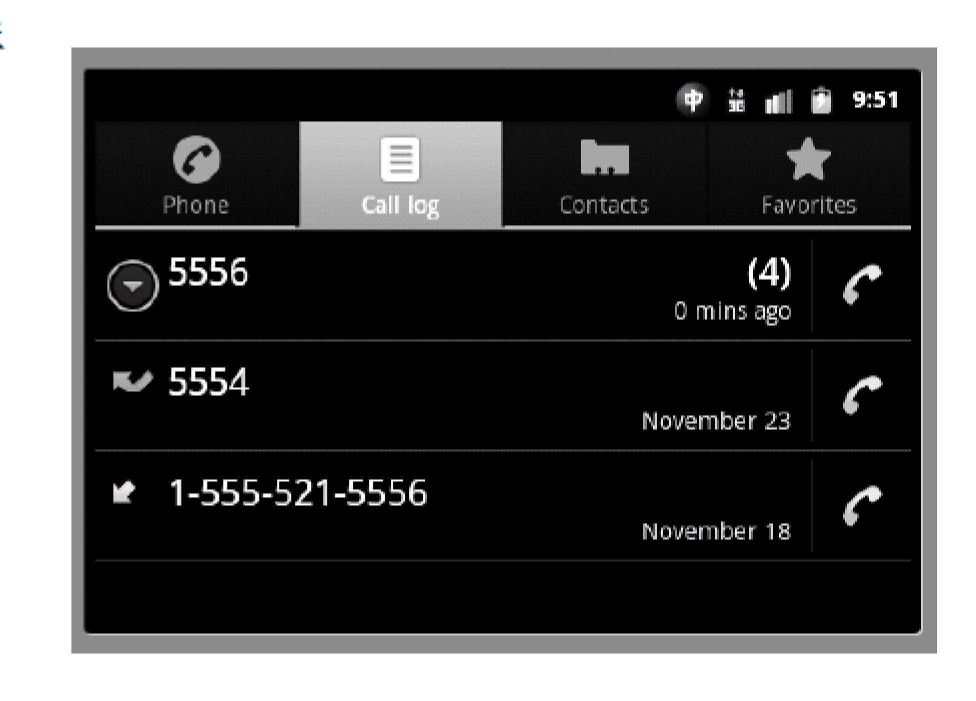


图 17-5-6 拨打电话

下面是输入电话号码之后再拨打电话的范例。

【例 17-5-8】 (代码位置:\17\testMakeIntent4. php)

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->makeIntent("android.
  intent.action.DIAL","tel:5556");
  $droid->startActivityIntent($ret['result']);
?>
```

程序运行后自动输入电话号码,但用户需要手工按呼叫按钮进行呼叫。运行结果如图 17-5-7 所示。

下面是发送 Intent 的范例，它用来发送“This is a text to send”，同时声明其内容为文本格式。

【例 17-5-9】 （代码位置：\17\testMakeInten5.php）

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->makeIntent("android.intent.action.SEND","This is a text to send","text/
  plain",$content);
  $droid->startActivityIntent($ret['result']);
?>
```

程序运行后会自动选用系统自带的 Messaging 程序发送文本内容，结果如图 17-5-8 所示。

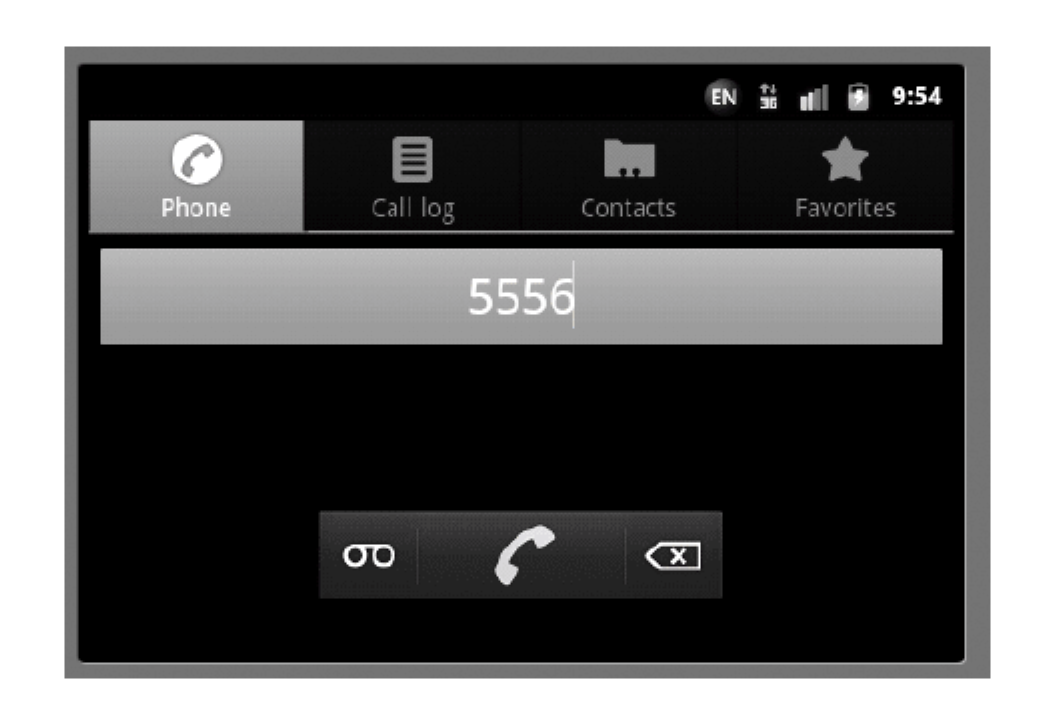


图 17-5-7 输入电话号码

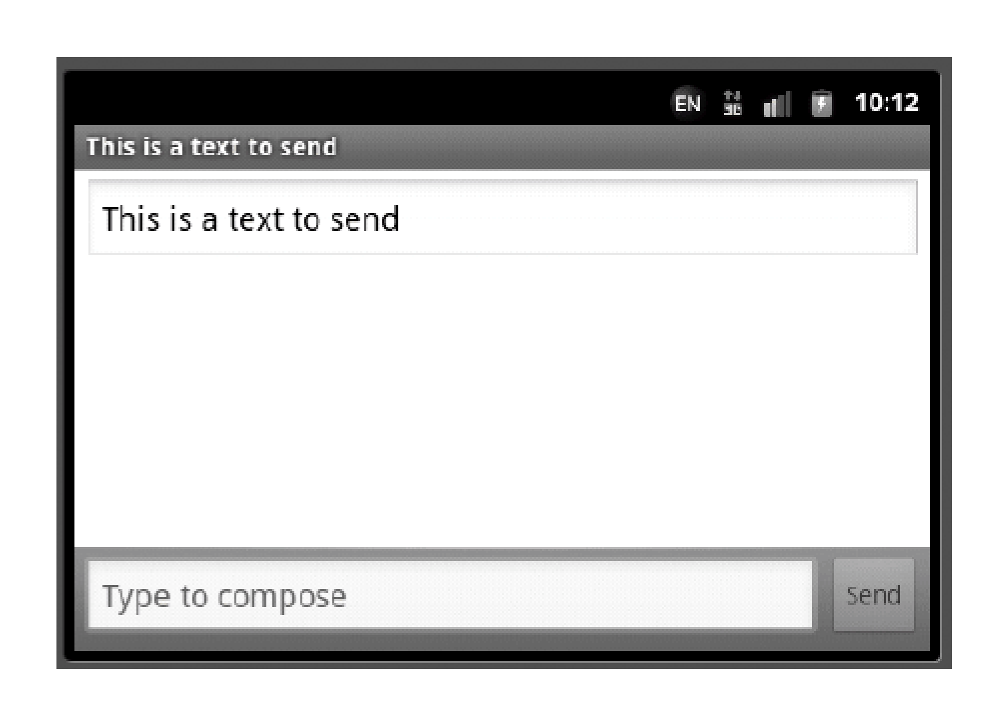


图 17-5-8 Intent 发送文本

17.5.7 广播

收听收音机是一种广播，在收音机中有很多个广播电台，每个广播电台播放的内容都不相同。广播方(发送方)并不在意接收方接收到广播时如何处理。收听交通电台的广播，电台会广播现在的交通状况，但它并不关心接收方接收到广播时如何做出处理，这不是广播应该关心的问题。在 Android 里面有各种各样的广播，比如电池的使用状态、电话的接收和短信的接收都会产生一个广播，应用程序也可以选择监听这些广播并进行处理。以手机来电为例，当手机来电时，应用程序接收到来电广播，之后程序会将数据库中的黑名单号码与来电号码做匹配，如果匹配的话则挂掉电话或静音处理。接收广播的程序是否会接收所有的广播？回答是否定的，程序只接收自己关心的广播，关心的广播数量可以是单个或多个。程序需要注册自己关心的广播，即告诉 Android 系统说“自己关心某个广播”，Android 系统遇到程序关心的广播就会通知程序，不是程序关心的广播是不会通知程序的。这样做的一个好处是性能会得到提高。下面介绍广播相关的函数。

```
sendBroadcastIntent(Intent intent)
```

函数 sendBroadcastIntent 用来发送广播，参数 intent 表示意图对象，可用函数 makeIntent 创建，它表示广播的内容。

```
eventRegisterForBroadcast(String category,Boolean enqueue)
```

函数 eventRegisterForBroadcast 用于注册广播。参数 category 表示广播类别(或名称),这个类别用字符串表示,是用户自定义的,其含义应该要明确;参数 enqueue 是布尔值,当值为 true 表示把广播加到消息队列中,否则进行分发,其默认值为 true,该参数是可选的。

```
eventWait(Integer timeout[optional]: the maximum time to wait)
```

函数 eventWait 用于等待事件的到来。参数 timeout 表示等待的最大时间,单位为毫秒,该参数可选,如果未指定该参数,则永远等待,直到有事件到来。函数会返回关联数组,数组包含了事件,返回的事件将从事件队列中删除。

下面是关于广播的范例,例中由文件 send. php 和 recv. php 组成,send. php 能让用户选择广播和发送广播,recv. php 注册了 3 个广播,然后等待接收广播和显示广播。

【例 17-5-10】 (代码位置:\17\Boardcast 目录)

文件 send. php 内容如下:

```
<?php
  require_once("Android.php");
  $droid=new Android();
  $droid->dialogCreateAlert("广播列表");
  $items=array();
  $items[0]='闹钟 1';
  $items[1]='闹钟 2';
  $droid->dialogSetItems($items);
  $droid->dialogShow();
  $response=$droid->dialogGetResponse();
  $droid->dialogSetItems(null);
  $item=$response["result"]->item+1;
  switch($item)
  {
      case 1:
              $ret=$droid->makeIntent("广播 1");
              $content=$ret['result'];
              $ret=$droid->sendBroadcastIntent($content);
              break;
      case 2:
              $ret=$droid->makeIntent("广播 2");
              $content=$ret['result'];
              $ret=$droid->sendBroadcastIntent($content);
              break;
      default:
               $ret=$droid->makeIntent("广播 3");
               $content=$ret['result'];
```

```
            $ret=$droid->sendBroadcastIntent($content);
            break;
    }
?>
```

文件 recv.php 内容如下：

```
<?php
  require_once("Android.php");
  $droid=new Android();
  //设置 3个广播监听器
  $droid->eventRegisterForBroadcast("广播 1");
  $droid->eventRegisterForBroadcast("广播 2");
  $droid->eventRegisterForBroadcast("广播 3");
  $ret=$droid->eventWait();        //最大时限地等待事件
  $name=$ret['result']->name;
  $data=$ret['result']->data;
  $droid->dialogCreateAlert("消息名:$name",$data);
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

先运行程序 recv.php，状态栏上将出现程序的图标，表明程序在运行中，但不会看到程序的运行界面，其实这个时候，该程序在等待事件的到来。然后运行程序 send.php，程序会弹出对话框提醒用户发送广播，用户可以任选一按钮发送广播。最后，程序 recv.php 跳出对话框显示广播，单击“确定”按钮后，会发现状态栏上的程序图标消失了，表明程序运行结束。图 17-5-9 是程序的运行结果，其中图 17-5-9(a)是程序 send.php 显示的广播列表，用于选择广播，在该次运行中选择了“闹钟 2”，图 17-5-9(b)是程序 recv.php 显示用户选择的广播。



(a) 广播列表



(b) 显示广播

图 17-5-9　广播

17.5.8 日志与调试

Android日志系统提供了记录和查看系统调试信息的功能。日志都是从各种软件和一些系统的缓冲区中记录下来的,缓冲区可以通过 logcat 命令或 DDMS 来查看和使用。每一个输出的 Android 日志信息都有一个标签和它的优先级,标签可以理解为某条日志记录的标题,用来简短地描述程序发生的事情,优先级从低到高分为 V、D、I、W 和 E,具体含义如下。

(1) V(verbose):显示全部信息。

(2) D(Debug):显示调试信息。

(3) I(Info):显示一般信息。

(4) W(Warning):显示警告信息。

(5) E(Error):显示错误信息。

可以通过函数 log 写信息到日志,再通过 DDMS 观察日志内容,这样就可以很方便地实现程序的调试。

```
log(String message)
```

log 函数用于将信息 message 写入 logcat。

下面是写日志的示例,例中会要求输入两个数以及计算两数之和,这两个数会写到日志中。

【例 17-5-11】 (代码位置:\17\testLog.php)

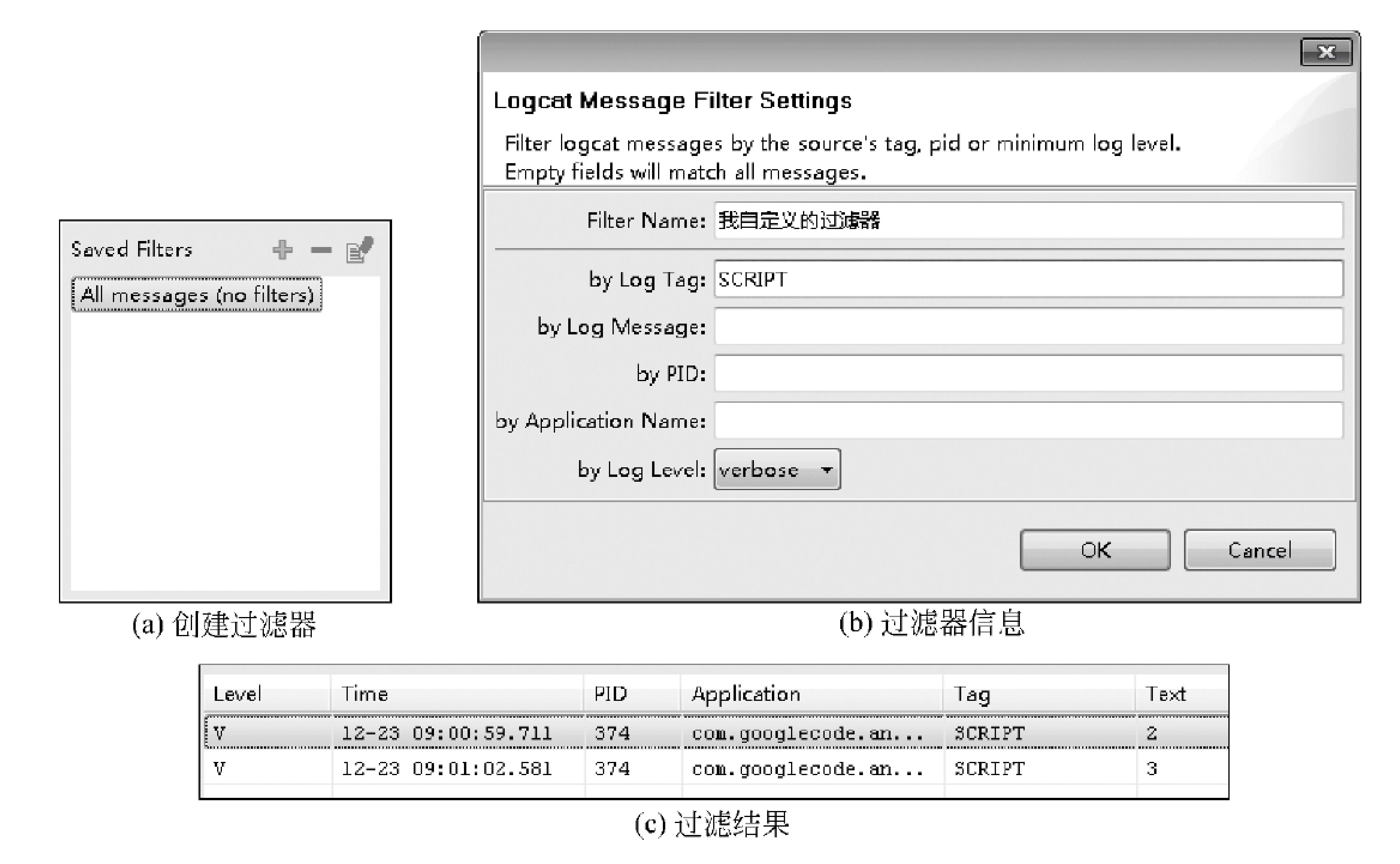
```
<?php
  require_once("Android.php");
  $droid=new Android();
  $ret=$droid->dialogGetInput("第一个数","请输入 0-9");
  $droid->dialogGetResponse();
  $droid->dialogDismiss();
  $a=$ret['result'];
  $droid->log($a);
  $ret=$droid->dialogGetInput("第二个数","请输入 0-9");
  $droid->dialogGetResponse();
  $droid->dialogDismiss();
  $b=$ret['result'];
  $droid->log($b);
  $c=$a+$b;
  $droid->dialogCreateAlert("两数之和","$c");
  $droid->dialogSetPositiveButtonText("确定");
  $droid->dialogSetNegativeButtonText("退出");
  $droid->dialogShow();
  $droid->dialogGetResponse();
?>
```

先运行程序 DDMS，然后再运行本程序并输入 2 和 3，最后使用 DDMS 观察日记是否记录了这两个数。会发现 DDMS 以表格的方式输出 2 和 3，其观察结果如图 17-5-10 所示，表格中的 Level 表示日志类别，Time 表示日志记录时间，PID 和 Application 分别表示进程号和应用程序，Tag 表示日志标签，PFA 会生成固定的标签 SCRIPT，Text 表示日志内容。

Level	Time	PID	Application	Tag	Text
V	12-23 09:00:59.571	374	com.googlecode.an...	sl4a.JsonRp...	Received: {"id":3,"method":"dialogDismiss","params":[]}
V	12-23 09:00:59.621	374	com.googlecode.an...	sl4a.JsonRp...	Sent: {"error":null,"id":3,"result":null}
V	12-23 09:00:59.671	374	com.googlecode.an...	sl4a.JsonRp...	Received: {"id":4,"method":"log","params":["2"]}
V	12-23 09:00:59.711	374	com.googlecode.an...	SCRIPT	2
V	12-23 09:00:59.711	374	com.googlecode.an...	sl4a.JsonRp...	Sent: {"error":null,"id":4,"result":null}
V	12-23 09:00:59.731	374	com.googlecode.an...	sl4a.JsonRp...	Received: {"id":5,"method":"dialogGetInput","params":["\u7b2c\u4e8c\u4e2a\u65

图 17-5-10 日志

如果日志数量太多，寻找日志记录会比较麻烦，可以通过日志过滤器筛选符合自定义条件的日志记录。首先需要在 DDMS 中创建过滤器，然后填写过滤器名称和过滤条件，可以通过标签、日志内容、日志优先级类别、PID 和应用程序名来过滤日志记录，最后使用过滤器过滤日志显示所需要的内容。过滤器创建和使用过程如图 17-5-11 所示，图 17-5-11(a)表示创建过滤器，图 17-5-11(b)是过滤器的名称和过滤条件，图 17-5-11(c)是使用过滤器过滤日志后的信息。



(a) 创建过滤器　(b) 过滤器信息

Level	Time	PID	Application	Tag	Text
V	12-23 09:00:59.711	374	com.googlecode.an...	SCRIPT	2
V	12-23 09:01:02.581	374	com.googlecode.an...	SCRIPT	3

(c) 过滤结果

图 17-5-11 日志过滤器

当在设备上进行调试时，由于 DDMS 并没有连接到设备，所以不能通过 DDMS 实时查看调试信息，那如何进行调试呢？可以先在手机端把信息写到日志中；之后在手机端设置 WiFi 使其能连接到网络；然后在手机端以 root 用户身份通过超级终端专业版 Better Terminal 设置 adb 连接端口，使 PC 端的 adb 程序能通过网络连接上手机设备，类

似的手机超级终端软件可以在 Android Market 上下载;最后在 PC 端使用 adb 的 logcat 命令查看日志内容。

在手机中打开 Better Terminal 终端(terminal)应用,输入命令设置 adb 连接端口,命令如图 17-5-12 所示。

可以根据用户的网络在首选项中设置 WiFi,关于 WiFi 设置可查看其他相关资料,设置连接后可以查看连接信息,如图 17-5-13 所示。

```
export PATH=/data/local/bin:$PATH:.
$ $su
# setprop service.adb.tcp.port 5555
# stop adbd
# start adbd
#
```

图 17-5-12　设置 adb 连接端口命令

状态消息	已连接
安全性	WPA/WPA2 PSK
信号强度	优良
连接速度	18Mbps
IP 地址	192.168.1.115

图 17-5-13　网络连接

在 PC 端控制台输入如下命令连接设备:

```
adb connect 192.168.1.115:5555
```

连接成功后,就可以在 PC 端控制台上输入以下命令查看日记:

```
adb shell logcat SCRIPT:V
```

logcat 是查看日志的命令,SCRIPT 是标签,V 是优先级,其意是查看标签为 SCRIPT 且优先级高于 V 的日志记录。运行之后可以查看到程序输出到日志的内容,部分运行结果如下:

```
V/SCRIPT  (  313): 2
V/SCRIPT  (  313): 3
```

17.5.9　振动

在吵闹的环境中,或者在不适宜使用视觉或听觉方式的情况下,振动是用来提醒用户的最佳方式。比如,在大街上或会议中收到短信或通知等,可以使用振动的方式通知用户有消息到来。振动也可以用在游戏当中加强游戏真实感,除此,振动还可以应用在闹钟中。振动功能必须在手机设备上进行测试,模拟器并不会以任何方式指示手机的振动,除此,某些 Android 手机并不支持振动功能。

```
vibrate(Integer duration[optional,default 300]: duration in milliseconds)
```

函数 vibrate 用于设置振动。参数 duration 表示振动时间,单位是 ms,默认是 300ms,该参数是可选的。

下面是振动的范例,例中会根据用户的选择来确定是否振动。

【例 17-5-12】　(代码位置:\17\testVibrate.php)

```
<?php
  require_once("Android.php");
```

```
  $droid=new Android();
  $droid->dialogCreateAlert("振动测试");
  $items=array();
  $items[0]='开始振动';
  $items[1]='取消振动';
  $droid->dialogSetItems($items);
  $droid->dialogShow();
  $droid->dialogSetItems(null);
  $response=$droid->dialogGetResponse();
  if($response["result"]->item==0)
    vibrate(1000);
?>
```

程序运行结果如图 17-5-14 所示，如果选择“开始振动”选项将振动手机 1s，如果选择“取消振动”将结束程序。需要注意的是，该程序必须在手机上运行，不能在模拟器和不支持振动功能的手机上运行。



图 17-5-14　振动测试

第 18 章



FullScreenUI 界面和事件处理

SL4A 支持 3 种方式实现界面：UiFacade、WebView 和 FullScreenUI。UiFacade 是指使用前面章节介绍的 Android 界面 API 实现对话框和进度条等界面，这种方式不能实现全屏界面，适合简单的应用。WebView 是指使用 HTML 设计实现 Android 界面，这种方法能实现复杂的全屏界面，但需要 WebKit 引擎解释 HTML 代码，占用资源较大。FullScreenUI 允许使用 XML 和 FullScreen API 设计实现界面，以及能对界面中的按钮单击和键盘事件进行响应，这种方法具有能实现复杂的全屏界面，占用资源小，使用 MVC 设计理念从而具有较好的扩展性等优点。本章主要介绍 FullScreenUI 设计实现界面，要使用这种方法实现界面，需要先掌握几个比较重要的函数，这些函数用来解析 XML 文件显示界面以及动态改变界面属性。

```
fullShow( String layout: String containing View layout, String title[optional]: Activity Title)
```

函数 fullShow 用来解析布局和显示界面，参数 layout 是布局，布局使用 XML 描述，参数 title 表示活动的标题，该参数可选。

```
fullDismiss()
```

函数 fullDismiss 用来隐藏当前界面。

```
fullQuery()
```

函数 fullQuery 用来查询界面的属性，返回值是界面属性集。

```
fullQueryDetail(String id)
```

函数 fullQueryDetail 用来查询指定控件的属性，参数 id 表示控件标识符，函数返回控件属性集。

```
fullSetProperty(String id,String property,String value)
```

函数 fullSetProperty 用来设置控件属性和属性值，参数 id 表示控件标识符，参数 property 表示控件的属性名，参数 value 表示属性值。

```
fullSetList(String id,JSONArray list)
```

函数 fullSetList 用来设置列表等控件的列表项,参数 id 表示列表等控件的标识符,参数 list 表示列表项,可用数组来表示。

```
fullKeyOverride(JSONArray keycodes,Boolean enable)
```

函数 fullKeyOverride 用来覆盖键盘事件,可用来改变系统的默认按键行为。参数 keycodes 表示要覆盖的按键值集。参数 enable 是布尔类型变量,用来表示是否打开或关闭默认按键行为,值 true 表示关闭按键默认行为,值 false 表示打开按键默认行为,该参数可选,默认值是 true。函数将返回原先设置的键值集。

18.1 界面布局管理

界面布局是指界面内容的组织结构,控件是界面的主要内容之一,布局决定了控件之间的结构。设计 Android 界面布局和 HTML 框架界面很相似,首先要给 Android 应用界面定框架,然后再在框架里面放控件,Android 系统提供了 AbsoluteLayout、LinearLayout、RelativeLayout、TableLayout 和 FrameLayout 布局。Android 提供了两种实现布局的方法,一种是在代码中实现界面布局,另一种是使用 XML 描述布局。使用 XML 描述布局是一种 MVC 实现方式,它使界面和逻辑分离,具有维护简便等优点。XML 描述布局适用于静态界面,它不能动态改变界面,而代码能够动态地改变界面。使用 XML 描述布局需要注意以下几点:

(1) 文件的扩展名必须是.xml。

(2) XML 的根节点通常是一个布局,布局里可以放置任意控件或布局子节点。

(3) 布局文件的根节点必须包含值为 http://schemas.android.com/apk/res/android 的命名空间。

(4) 如果需要在代码中使用控件,必须给控件定义一个 id,一个新 id 的定义格式为 @+id/myId。

(5) 控件的大小和位置通常由布局来控制,不同的布局有不同的方法,一般会使用两个属性值 wrap_content 和 fill_parent(或 match_parent)来决定,前者让控件根据自己的内容来定义大小,后者让控件充满父控件。

(6) 属性 android:gravity 和 android:layout_gravity 都用于对齐,android:gravity 用于设置控件的对齐方式,而 android:layout_gravity 用于设置它在父控件中的对齐方式。例如,一个按钮中的文字的对齐方式要使用属性 android:gravity 设置,而控件的位置对齐方式则使用 android:layout_gravity 设置。

(7) 设置宽高等距离时要指定单位,单位可以是 px(像素)、dip(设备独立像素)和 sp(比例像素)等。

18.1.1 LinearLayout 布局

LinearLayout 也称线性布局,线性布局又分垂直线性布局和水平线性布局。垂直线性布局是单列多行结构,每行只能放置一个控件,控件依照从顶行到底行顺序放置;水平

线性布局是单行多列结构，每列只能放置一个控件，控件依照从左列到右列顺序放置。布局是可以嵌套的，即布局里可以含有布局。如果搭建两行两列的布局，那么通常的方式是先垂直排列两个元素，每一个元素里再包含一个LinearLayout进行水平排列。

LinearLayout中的子元素属性android:layout_weight将生效，其用于描述该子元素在剩余空间中占有的大小比例。如果一行只有一个文本框，那么它的默认值就为0。如果一行中有两个等长的文本框，那么它们的android:layout_weight值可以是同为1。如果一行中有两个不等长的文本框，那么它们的android:layout_weight值分别为1和2，第一个文本框将占据剩余空间的三分之二，第二个文本框将占据剩余空间中的三分之一。android:layout_weight遵循数值越小其重要度越高的原则。下面是线性布局的范例，由两个文件组成，一个是XML文件，用于描述界面布局，另一个PHP文件读取XML文件并交由系统处理并显示出来。

【例18-1-1】（代码位置：\18\linearlayout目录）

文件mylayout.xml内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <TextView android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:background="#000000"
            android:text="@string/hello"/>
  <LinearLayout android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:background="#654321"
             android:layout_weight="1" android:text="1"/>
    <TextView android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:background="#fedcba"
             android:layout_weight="2" android:text="2"/>
  </LinearLayout>
</LinearLayout>
```

文件test.php内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$droid->eventWait(5000);
?>
```



图 18-1-1　线性布局

程序使用了一个垂直线性布局，布局里包含了一个控件和一个水平线性布局，水平线性布局里包含有两个控件。先把文件 mylayout.xml 复制到模拟器中的/sdcard/sl4a/scripts/目录，然后再运行程序，运行结果如图 18-1-1 所示。

18.1.2　FrameLayout 布局

FrameLayout 也称框架布局，FrameLayout 是五大布局中最简单的一个布局，在这个布局中，所有的子元素以层叠的方式放置在同一个区域中，第一个添加的元素放在最低层，最后添加的放在最上面，下面的元素将被上面的元素覆盖。下面是框架布局的范例，其显示效果是：第一个 TextView 被第二个 TextView 完全遮挡，第三个 TextView 遮挡了第二个 TextView 的部分位置。

【例 18-1-2】（代码位置：\18\framelayout 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="fill_parent"
  android:layout_height="fill_parent">
   <TextView android:layout_width="fill_parent" android:layout_height="fill_parent"
      android:background="#000000" android:gravity="center" android:text="1"/>
   <TextView android:layout_width="fill_parent" android:layout_height="fill_parent"
      android:background="#654321" android:gravity="center" android:text="2"/>
   <TextView android:layout_width="50dp" android:layout_height="50dp"
      android:background="#fedcba" android:gravity="center" android:text="3"/>
</FrameLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout = file _ get _ contents ("/sdcard/sl4a/scripts/
mylayout.xml");
$droid->fullShow($layout);
$droid->eventWait(5000);
?>
```

先把文件 mylayout.xml 复制到模拟器中的/sdcard/sl4a/scripts/目录，然后再运行程序，运行结果如图 18-1-2 所示。

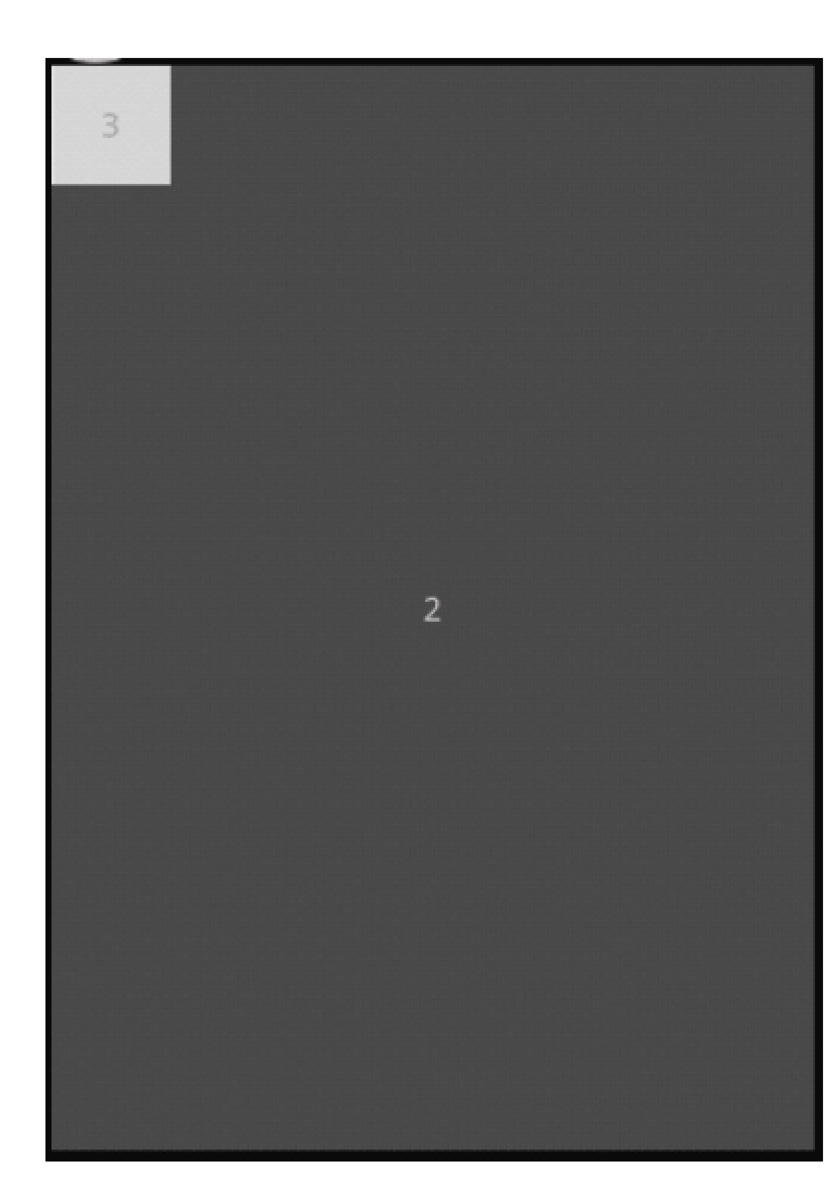


图 18-1-2　框架布局

18.1.3 AbsoluteLayout 布局

AbsoluteLayout 也称绝对布局，在此布局中的子元素的属性 android:layout_x 和 android:layout_y 将生效，用于描述该子元素的坐标位置。屏幕左上角为坐标原点(0,0)，第一个 0 代表横坐标，向右移动此值增大；第二个 0 代表纵坐标，向下移动，此值增大。在此布局中的子元素可以相互重叠。在实际开发中，通常不采用此布局格式，因为不同的设备的分辨率不同，绝对布局在不同的设备上运行的效果往往不理想，甚至有可能不能很好地适配各种终端。

18.1.4 RelativeLayout 布局

RelativeLayout 也称相对布局，RelativeLayout 按照各子元素之间的位置关系完成布局，它允许子控件放置在与父控件或其他子控件相对的位置。在此布局中的子元素里与位置相关的属性将生效，例如 android:layout_below 和 android:layout_above 等。子元素使用这些位置属性时需要指定被参照的控件，可以发现这些属性的值的格式都为 @id/idname。注意，在指定位置关系时，引用的 ID 必须在引用之前先被定义，否则将出现异常。RelativeLayout 常用的位置属性如表 18-1-1 所示。

表 18-1-1 RelativeLayout 常用属性

属　　性	说　　明
android:layout_toLeftOf	控件位于引用控件的左方
android:layout_toRightOf	控件位于引用控件的右方
android:layout_above	控件位于引用控件的上方
android:layout_below	控件位于引用控件的下方
android:layout_alignParentLeft	控件是否与父控件的左端对齐
android:layout_alignParentRight	控件是否与父控件的右端对齐
android:layout_alignParentTop	控件是否与父控件的顶部对齐
android:layout_alignParentBottom	控件是否与父控件的底部对齐
android:layout_centerInParent	控件是否相对于父控件居中对齐
android:layout_centerHorizontal	控件是否水平居中
android:layout_centerVertical	控件是否垂直居中

RelativeLayout 是 Android 五大布局结构中最灵活的一种布局结构，比较适合一些复杂界面的布局。下面是相对布局的范例，第一个控件与父控件的底部对齐，第二个控件位于第一个控件的上方，并且第三个控件位于第二个控件的左方。

【例 18-1-3】 （代码位置：\18\relativelayout 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent">
    <TextView android:id="@+id/text_01" android:layout_width="50dp"
      android:layout_height="50dp" android:background="#ffffff"
      android:gravity="center" android:layout_alignParentBottom="true"
      android:text="1"/>
    <TextView android:id="@+id/text_02"
      android:layout_width="50dp" android:layout_height="50dp"
      android:background="#654321" android:gravity="center"
      android:layout_above="@id/text_01"
      android:layout_centerHorizontal="true" android:text="2"/>
    <TextView android:id="@+id/text_03" android:layout_width="50dp"
      android:layout_height="50dp" android:background="#fedcba"
      android:gravity="center" android:layout_toLeftOf="@id/text_02"
      android:layout_above="@id/text_01" android:text="3"/>
</RelativeLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$droid->eventWait(5000);
?>
```

先把文件 mylayout.xml 复制到模拟器中的/sdcard/sl4a/scripts/目录，然后再运行程序，运行结果如图 18-1-3 所示。



图 18-1-3 相对布局

18.1.5 TableLayout 布局

TableLayout 也称表格布局，这种布局是把元素放到表格中的行或列中，每个表格可以由多个 TableRow 来定义行，一个 TableRow 代表一行，TableLayout 的列数等于含有最多子控件的 TableRow 的列数，值得注意的是，布局并不会显示表格的边框。表格中的单元格可以为空，控件可以根据需要放置到指定列中，控件也可以占用多列。表格布局常用属性和单元格属性如表 18-1-2 所示。

表 18-1-2 表格布局和单元格属性

属　性	说　明
android:layout_column	将控件放置到指定列位置，列的开始位置从 0 开始计算，例如，android:layout_column＝"1"表示该控件显示在第 1 列
android:layout_span	指定控件占据的列数，例如，android:layout_span＝"2"表示该控件占据 2 列。如未指定该属性则占 1 列

续表

属　　性	说　　明
android:collapseColumns	设置要隐藏的列
android:shrinkColumns	设置可收缩的列。当该列子控件的内容太多,已经挤满所在行时,该子控件的内容将往列方向显示
android:stretchColumns	设置可伸展的列。该列可以向行方向伸展,最多可占据一整行

下面的示例演示了一个 TableLayout 的布局结构,该布局是 3 行 3 列的表格结构,第一行包含 2 个控件,其中有一个控件占有 2 个单元格;第二行含有 2 个控件,其中第一个控件被指定放置到了第 1 列,第二个控件会顺序被放置到第 2 列,第 0 列没有控件;第三行有 3 个控件,这 3 个控件并没有顺序放置,而是按指定位置放置。

【例 18-1-4】 (代码位置:\18\tablelayout 目录)

文件 mylayout.xml 内容如下:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent" android:layout_height="fill_parent"
>
   <TableRow android:layout_width="fill_parent" android:layout_height=
     "wrap_content">
      <TextView android:background="#ffffff" android:gravity="center"
      android:padding="10dp" android:text="1"/>
      <TextView android:background="#654321" android:gravity="center"
      android:padding="10dp" android:text="2" android:layout_span="2"/>
   </TableRow>
   <TableRow android:layout_width="fill_parent" android:layout_height=
     "wrap_content">
      <TextView android:background="#654321" android:gravity="center"
      android:padding="10dp" android:text="2" android:layout_column="1"/>
      <TextView android:background="#fedcba" android:gravity="center"
      android:padding="10dp" android:text="3"/>
   </TableRow>
   <TableRow android:layout_width="fill_parent" android:layout_height=
     "wrap_content">
      <TextView android:background="#fedcba" android:gravity="center"
      android:padding="10dp" android:text="3"/>
      <TextView android:background="#654321" android:gravity="center"
      android:padding="10dp" android:text="2"/>
      <TextView android:background="#ffffff" android:gravity="center"
      android:padding="10dp" android:text="1"/>
   </TableRow>
</TableLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/
scripts/mylayout.xml");
$droid->fullShow($layout);
$droid->eventWait(5000);
?>
```

先把文件 mylayout.xml 复制到模拟器中的/sdcard/sl4a/scripts/目录，然后再运行程序，运行结果如图 18-1-4 所示。

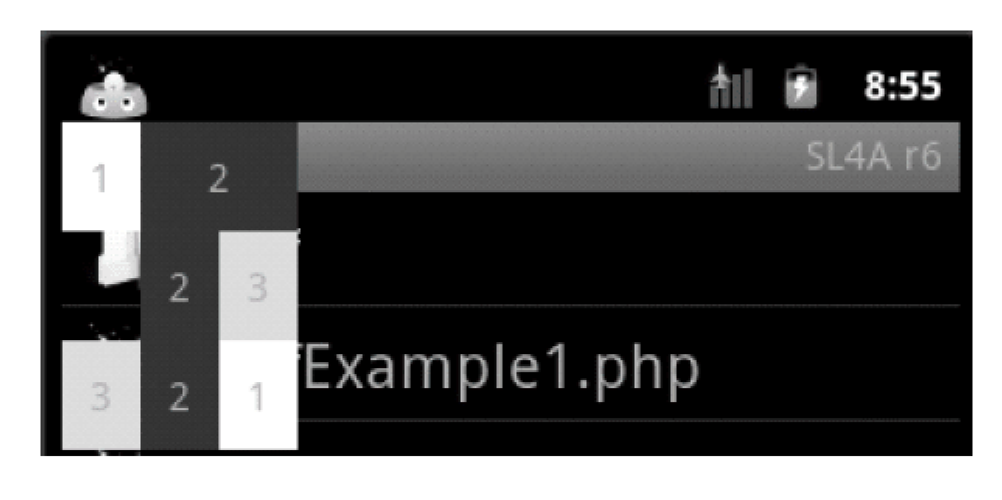


图 18-1-4　表格布局

18.2　界面控件

18.2.1　TextView

TextView 控件一般用来显示固定长度的文本或者标签，控件可对文本的颜色和大小等属性进行设置，控件常用属性如表 18-2-1 所示。

表 18-2-1　TextView 属性

属　性	说　明
android:gravity	设置控件文字对齐方向，其值为 left、center 或 right
android:height	设置控件高度
android:width	设置控件宽度
android:hint	控件 text 属性内容为空时显示的文本(提示信息)
android:textColorHint	设置提示信息文字的颜色，默认为灰色，与 hint 一起使用
android:text	设置文本内容
android:textColor	设置文本颜色
android:textSize	设置文本字体大小
android:typeface	文本字体，Android 系统默认支持 3 种字体，分别为 sans、serif 和 monospace，除此之外还可以使用其他字体文件(*.ttf)
android:background	设置背景颜色
android:textStyle	设置字体为粗体和斜体，取值为 bold(粗体)和 italic(斜体)。值可以组合，组合时值之间要加 | 符号。例如，android:textStyle="bold|italic"表示文本为粗斜体
android:lines	设置文本的行数，设置两行就显示两行，即使第二行没有数据

Android 支持 24 位 RGB 颜色和包含透明度的 24 位 RGB 颜色，颜色可使用十六进制数来表示。对于包含透明度的 24 位 RGB 颜色，前面 2 位十六进制数表示透明度，值越小越透明，值越大越不透明。例如，#ff0000 表示红色，#80ff0000 表示半透明的红色，#ffff0000 表示不透明的红色。下面是 TextView 的范例。

【例 18-2-1】 （代码位置：\18\textview 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <TextView
    android:text="这是 TextView 范例,本范例的文字居中对齐。"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25px"
    android:textColor="#fff00000"
    android:background="#00ff00"
    android:textStyle="bold"
    android:gravity="center"
    android:height="200px"
  />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$droid->eventWait(5000);
?>
```

程序运行结果如图 18-2-1 所示。

注意：本节后面的范例如不做说明，将只给出文件 mylayout.xml 而不再给出文件 test.php，因为 test.php 内容相同，同时，运行程序前需把文件 mylayout.xml 和 test.php 复制到手机或模拟器的/sdcard/sl4a/scripts 目录。



图 18-2-1 TextView

18.2.2 EditText

EditText 是文本编辑框控件，它是一个非常重要的组件，通常用来接收用户的输入

和把用户的输入传输给 Android 应用进行处理。用户的输入通常都是有一定格式的，例如，用户输入的内容可以是电话号码、日期和网址等，在开发的过程中，通常都会在 XML 文件中加入 android:inputType 属性和设置相应的属性值使 EditText 控件只接收指定格式的数据，同时，EditText 控件还会让虚拟键盘适应输入框中内容的类型。属性 android:inputType 的值和值含义如表 18-2-2 所示。

表 18-2-2 android:inputType 值含义

值	含 义	值	含 义
text	任何文本	date	日期
number	数字	textMultiLine	多行输入
textEmailAddress	邮箱地址格式	singleLine	单行输入
textUri	网址	textPassword	密码输入
phone	电话号码	datetime	时间日期
time	时间	numberDecimal	带小数点的数字

下面是文本框的范例，例中文本框可接受普通文本，没有格式约束。

【例 18-2-2】 （代码位置：\18\edittext 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000" android:gravity="center">
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="text"
  />
</LinearLayout>
```

123abcABC@#$

图 18-2-2 文本框

程序运行后输入字符，其运行结果如图 18-2-2 所示。

下面是密码框的范例，当用户在密码框中输入文本时，密码框将不显示文本内容而是用点表示文本内容。

【例 18-2-3】 （代码位置：\18\pwdtext 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000" android:gravity="center">
```

```
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="textPassword"
  />
</LinearLayout>
```

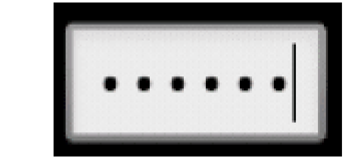


图 18-2-3　密码框

程序运行后输入密码，其运行结果如图 18-2-3 所示。

下面是多行文本框的范例，当文本框一行文本满屏时会自动换行，文本框也接受回车换行符。

【例 18-2-4】（代码位置：\18\multilinetext 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent" android:
background="#ff000000" android:gravity="center">
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="textMultiLine"
  />
</LinearLayout>
```

图 18-2-4　多行文本框

程序运行后输入文本，其结果如图 18-2-4 所示。

下面是电话格式文本框的范例，当在文本框中输入信息时，系统会弹出软键盘供用户输入，软键盘只提供拨打电话时所涉及的符号，这样可以避免误输入。值得注意的是，其他格式的文本框也会提供软键盘，但会根据输入信息的格式提供不同的软键盘，例如，textUri 和 Date 格式的文本框会提供网址和日期格式的软键盘。

【例 18-2-5】（代码位置：\18\phonetext 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000" android:gravity="center">
  <EditText
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
```

```
    android:textSize="20px"
    android:inputType="phone"
    android:text="12345678"
  />
</LinearLayout>
```

程序运行后输入信息，其运行结果如图 18-2-5 所示。

图 18-2-5 电话输入框

18.2.3 Button

Button 是 Android 按钮控件，是比较常用的控件，用户可以单击它实现各种操作，例如，单击按钮实现视频播放暂停或播放等操作。Button 控件可以通过设置属性和属性值来改变按钮控件外观，例如，可以改变按钮控件的字体和颜色等。下例是按钮控件的范例。

【例 18-2-6】 （代码位置：\18\button 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <Button
      android:text="请点击这里"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/mybutton"
      android:textStyle="bold"
      android:textSize="20px"
      >
  </Button>
</LinearLayout>
```



图 18-2-6 按钮

程序运行结果如图 18-2-6 所示。

18.2.4 ImageButton

Android 提供了另一种按钮控件，这种按钮控件以图片作为整个按钮，这就是 ImageButton，它可以使按钮外观更美观并能实现自定义风格的按钮。ImageButton 引用的图片可以通过 android:src 属性来定义，它有两种定义格式，具体如表 18-2-3 所示。

表 18-2-3 android:src 格式及含义

格　　式	含　　义
android:src="@android:drawable/资源名"	引用系统提供的标准图片资源，例如，android:src="@android:drawable/stat_sys_phone_call_on_hold"
android:src=" file:///目录和文件名"	引用图像文件资源，例如，android:src =" file:///sdcard/download/panda72.png"

ImageButton 能以文件 URL 的形式引用图像资源，支持的图像文件格式是 png 和 jpg 格式，或者通过指定资源名引用已编译到 SL4A 中的资源和 Android 系统提供的标准资源，这些标准资源是所有应用程序共享的。关于可用系统资源的完整列表请参阅 http://developer.android.com/reference/android/R.html。除了图像资源外，Android 还提供了很多类型的资源，包括字符串和图标等，可通过下面格式来引用资源。

```
@包名:资源类型/资源名
```

下例是使用系统标准资源的范例。

【例 18-2-7】 （代码位置：\18\imagebutton1 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000">
  <ImageButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/mybutton"
      android:src="@android:drawable/stat_sys_phone_call_on_hold"
      >
  </ImageButton>
</LinearLayout>
```


图 18-2-7 ImageButton

程序运行结果如图 18-2-7 所示。

下面是引用图像文件的范例。

【例 18-2-8】 （代码位置：\18\imagebutton2 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#000000">
  <ImageButton
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/mybutton"
        android:src="file:///sdcard/sl4a/scripts/moon.png"
        >
    </ImageButton>
</LinearLayout>
```

先把图像上传到模拟器的指定目录，然后再运行程序，运行后会显示图像，这个图像就是按钮，可以像普通按钮一样被单击。运行结果如图 18-2-8 所示。



图 18-2-8　引用图像文件的按钮

18.2.5　RadioButton 和 CheckBox

有时候应用会提供一组选项供用户选择，用户能从中选择一项而且只能选择一项，例如，男女选项和身体健康状态选项(健康、亚健康和不健康 3 个选项)。可以借助 RadioButton 和 RadioGroup 标签来实现这种功能，RadioButton 称作单选按钮控件，RadioGroup 称作单选组合框，RadioGroup 用于将 RadioButton 框起来形成一组单选项供用户选择。在没有 RadioGroup 的情况下，RadioButton 可以全部都选中；当多个 RadioButton 被 RadioGroup 包含的情况下，RadioButton 只可以选择一个。

下面是 RadioButton 和 RadionGroup 的范例。

【例 18-2-9】　(代码位置：\18\radion 目录)

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000"
android:textSize="25px"
>
  <TextView
    android:text="请问你是美女还是帅哥?"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
  />
  <RadioGroup
      android:orientation="horizontal"
      android:layout_width="fill_parent"
      android:layout_height="match_parent">
    <RadioButton
```

```
            android:text="美女"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/girl">
        </RadioButton>
        <RadioButton
            android:text="帅哥"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/boy">
        </RadioButton>
    </RadioGroup>
</LinearLayout>
```

程序运行结果如图 18-2-9 所示。

请问你是美女还是帅哥？
美女 帅哥

图 18-2-9 单选按钮

有时候应用会提供一组选项供用户选择，用户可以从这组选项中选若干个选项。例如，实现一道简单的多选题，用户可根据情况选中一个或多个选项；再比如，调查兴趣应用会提供多个兴趣选项供用户选择，用户根据自己的兴趣可以选择多个选项。这种应用可以使用 CheckBox 控件实现，CheckBox 控件也称为复选框控件，CheckBox 控件没有分组概念。下面是多选应用的范例。

【例 18-2-10】 （代码位置：\18\check 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#000000"
>
    <CheckBox
        android:id="@+id/mycheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="是否喜欢唱歌?">
    </CheckBox>
    <CheckBox
        android:id="@+id/mycheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="是否喜欢打篮球?">
    </CheckBox>
    <CheckBox
        android:id="@+id/mycheckBox"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="是否喜欢游泳?">
    </CheckBox>
    <Button
        android:text="提交答案"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/mybutton"
        android:textStyle="bold"
        android:textSize="20px"
        >
    </Button>
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$ret1=$droid->eventWait(12000);
while($ret1['result']->data->id!='mybutton')
    $ret1=$droid->eventWait(12000);
?>
```

程序运行后，用户可以选择多个选项，运行结果如图 18-2-10 所示。

是否喜欢唱歌？
是否喜欢打篮球？
是否喜欢游泳？
提交答案

图 18-2-10　复选框

18.2.6　ToggleButton

无论复选框是否被选中，其旁边显示的文字并不会发生变化，如果要让旁边的文字发生变化，就可以使用 ToggleButton 控件来取代复选框控件来实现这一功能。ToggleButton 控件称开关按钮控件，它有开与关两种状态，通过 android:textOn 和 android:textOff 属性可以设置开与关状态下要显示的文字，android:isChecked 属性可用来判断开关按钮的状态，通过 android:checked 属性可以设置开关按钮的状态。

下面是开关按钮的范例。

【例 18-2-11】（代码位置：\18\togglebutton 目录）

文件 mylayout.xml 内容如下：

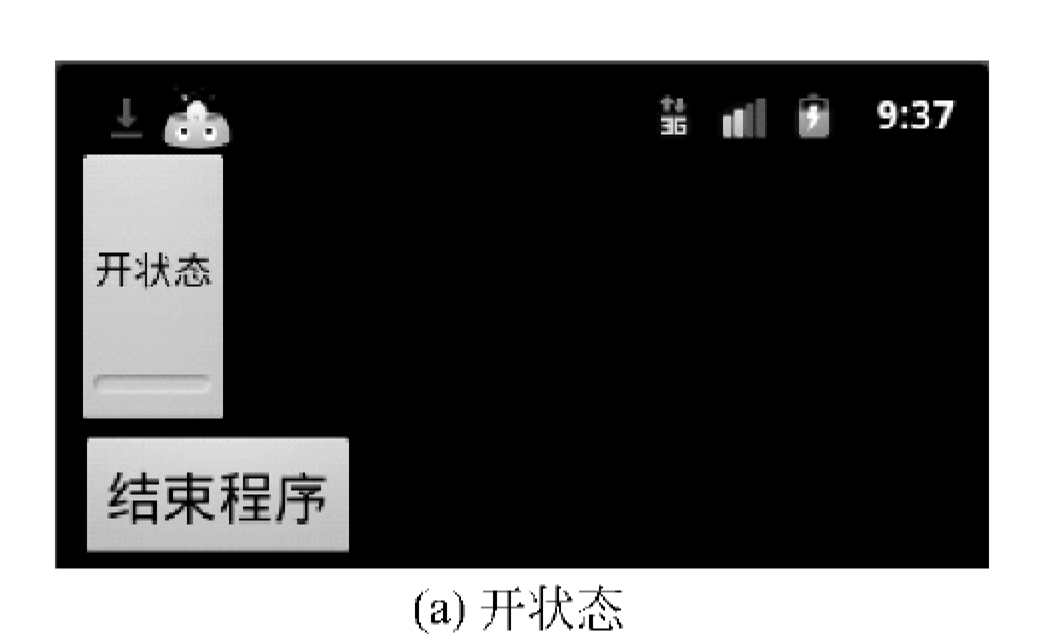
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000"
```

```
>
  <ToggleButton
      android:id="@+id/mytogglebutton"
      android:layout_width="wrap_content"
      android:layout_height="100px"
      android:textOn="开状态"
      android:textOff="关状态"
      android:checked="true">
  </ToggleButton>
  <Button
      android:text="结束程序"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/mybutton"
      android:textStyle="bold"
      android:textSize="20px"
      >
  </Button>
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$ret1=$droid->eventWait(12000);
while($ret1['result']->data->id!='mybutton')
    $ret1=$droid->eventWait(12000);
?>
```

程序运行后会显示两个按钮，一个是开关按钮，另一个是普通按钮，当单击开关按钮时，开关按钮上的文字会发生变化，当单击普通按钮时，程序会结束。程序运行结果如图 18-2-11 所示，图 18-2-11(a)显示开状态，图 18-2-11(b)显示关状态。



(a) 开状态



(b) 关状态

图 18-2-11 开关按钮

18.2.7 DatePicker和TimePicker

许多应用都需要用户输入日期和时间，例如网络订票、网络订房和注册会员。有些应用提供了万年历让用户选择日期和时间而不需要手工输入，这极大地方便了用户。Android提供了DatePicker和TimePicker控件实现这种功能以，DatePicker控件称为日期选择控件，通过日期选择控件能够快速选择日期，日期包含年月日数据。TimePicer控件称为时间选择控件，时间选择控件向用户显示一天中的时间(可以为24小时，也可以为AM/PM制)，并允许用户进行选择。下面是日期和时间选择的范例。

【例18-2-12】（代码位置：\18\dtpicker目录）

文件mylayout.xml内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000000">
  <TextView
    android:text="请选择日期和时间"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
  />
  <DatePicker android:id="@+id/datePicker"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_horizontal"
      />
  <TimePicker android:id="@+id/timePicker"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_horizontal"
      />
  <Button
    android:text="设置"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/mybutton"
    android:textStyle="bold"
    android:textSize="20px"
    >
  </Button>
</LinearLayout>
```

文件 test. php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$ret1=$droid->eventWait(12000);
while($ret1['result']->data->id!='mybutton')
    $ret1=$droid->eventWait(12000);
?>
```

程序运行后，单击“＋”或“－”按钮修改日期和时间，但此时并没有发生变化，因为其并没有响应事件，关于事件在后面章节介绍。要退出程序，可以单击“设置”按钮。运行结果如图 18-2-12 所示。



图 18-2-12　日期和时间

18.2.8　ImageView

Android 提供了 ImageView 控件用来显示图像，它可以加载系统标准资源和图片文件资源，它可以指定修改原图的尺寸。下面是 ImageView 的范例，例中显示两个图像，其中一个是系统标准图像，另一个是文件图像，例中通过 android:layout_height 和 android:layout_width 指定了图像的高度和宽度。

【例 18-2-13】（代码位置：\18\imageview 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent" android:background="#ff000000">
  <ImageView
    android:src="@android:drawable/stat_sys_phone_call_on_hold"
    android:layout_width="100px"
    android:layout_height="100px"
    >
  </ImageView>
  <ImageView
    android:src="file:///sdcard/sl4a/scripts/moon.png"
    android:layout_width="150px"
    android:layout_height="150px"
    >
```

```
  </ImageView>
</LinearLayout>
```

先把图像文件 moon. png 复制到指定目录，然后运行程序显示图像，其中电话图像是系统标准图像资源，其运行结果如图 18-2-13 所示。

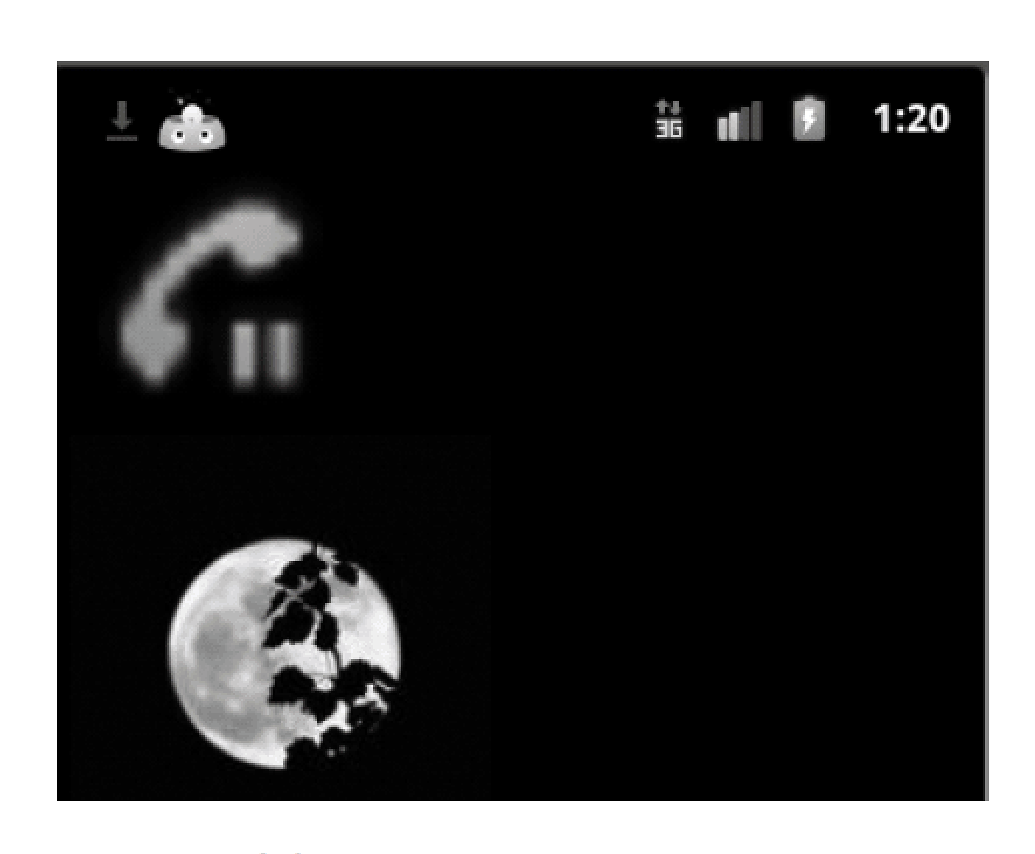


图 18-2-13　ImageView

18.2.9 ProgressBar、SeekBar 和 RatingBar 控件

有些操作比较耗时，例如文件下载，为了让用户看到操作的进度，Android 提供了 ProgressBar(进度条)控件。ProgressBar 控件有两个进度条，分别称为第一层进度条和第二层进度条，第一层进度条为用户呈现操作的进度，第二层用来显示中间进度。例如，在播放流媒体视频时，第一层进度条显示多媒体播放进度，第二层进度条显示缓冲区进度。默认的进度条常用于应用程序任务完成时间不确定的情景。PHP for Android 对使用 XML 描述 ProgressBar 进度条的支持不够，ProgressBar 控件的使用可参看前面章节的介绍。

SeekBar 控件称为拖动进度条，它可以通过滑块的位置来标识进度，而且拖动条允许用户拖动滑块来改变值，因此拖动进度条通常用于对系统的某种数值进行调节，比如调节音量等。拖动进度条比较常用的属性如表 18-2-4 所示。

表 18-2-4　SeekBar 常用属性

属　　性	含　　义
android:max	拖动条最大值，例如，android:max="100"
android:progress	拖动条当前值，例如，android:progress="10"
android:secondaryProgress	第二层进度条的进度值，例如，android:secondaryProgress="70"

下面是 SeekBar 控件的范例。

【例 18-2-14】（代码位置：\18\seekbar 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#ff000000"
    >
  <SeekBar
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_marginBottom="10dp"
```

```
        android:secondaryProgress="70"
        android:max="100"
        android:progress="50"
    />
    <SeekBar
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:secondaryProgress="80"
        android:max="100"
        android:progress="20"
    />
</LinearLayout>
```

程序运行结果如图 18-2-14 所示。

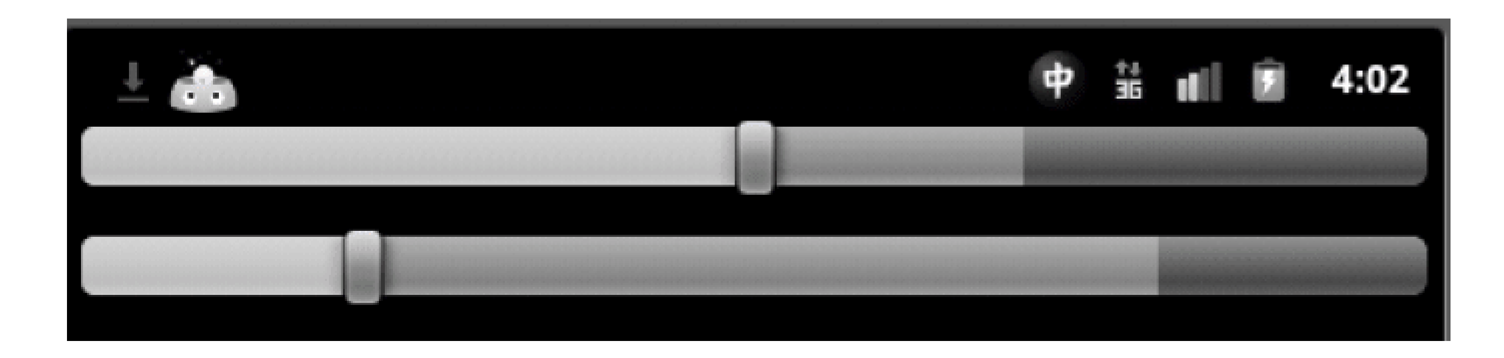


图 18-2-14　SeekBar 控件

RatingBar 也叫评分条，是 SeekBar 和 ProgressBar 的扩展，用星星来评级。通过评分条，用户可以触摸/拖动或使用键来设置评分，这种评分条默认有 5 个星图案，可以通过属性改变星星的数量和水平拖动步长等方面的内容，常见的属性如表 18-2-5 所示。

表 18-2-5　评分条属性

属　　性	含　　义
android:isIndicator	值 true 表示评分条是一个指示器(用户无法拖动星星图案进行评分)
android:numStars	显示的星星数量，必须是一个整形值，例如，android:numStars="8"
android:rating	默认的评分，是浮点类型，例如，android:rating="1.2"
android:stepSize	评分的步长，是浮点类型，例如，android:stepSize="1.2"

下面是评分条的范例，例中有 3 个评分条。第一个是指示器，有 5 个星星图案，它只能用显示星星图案而不能拖动星星图案。第二个是使用默认属性值的评分条，默认的评分条有 5 个星星图案，使用默认的拖动步长。第三个评分条有 4 个星星图案，初始有 2 个星星图案被选中，水平拖动步长是 0.1。

【例 18-2-15】（代码位置：\18\ratingbar 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent"
    android:orientation="vertical" >
  <RatingBar
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      style="?android:attr/ratingBarStyleIndicator"
      android:numStars="4"
      android:rating="2.5"
      android:stepSize="0.1"
      />
  <RatingBar
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      style="?android:attr/ratingBarStyleIndicator"
      />
  <RatingBar
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      style="?android:attr/ratingBarStyleIndicator"
      android:isIndicator="true"
      />
</LinearLayout>
```

程序运行后，因为最下面的评分条的属性 android:isIndicator 值为真，所以它是指示器，用户不能拖动星星图案，而最上面的两个评分条的星星图案是可以拖动的，但这两个评分条的拖动步长不同，一个使用默认的拖动步长，而另一个使用的拖动步长是 0.1。运行结果如图 18-2-15 所示。

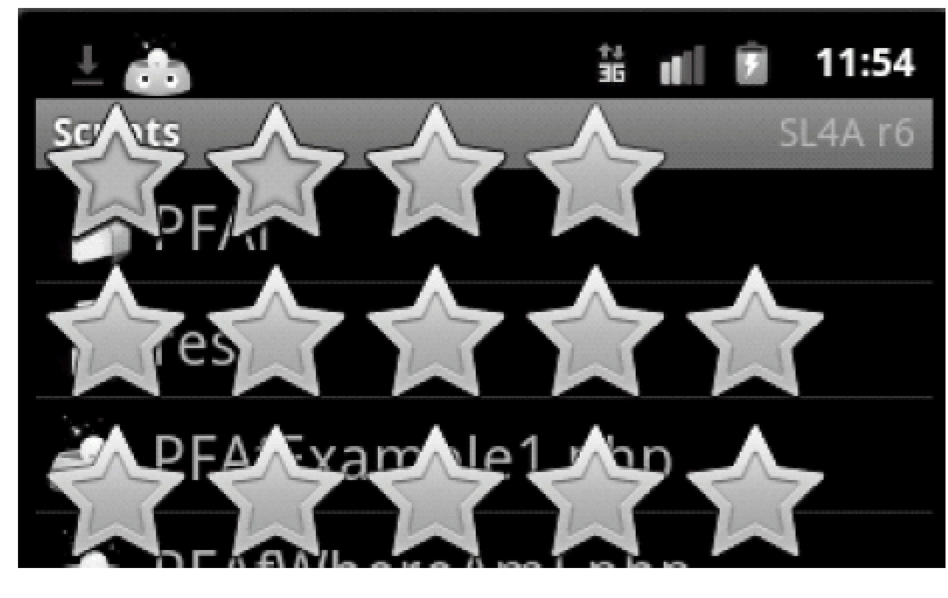


图 18-2-15　评分条

18.2.10　Spinner 和 ListView 控件

Spinner 控件是 Android 系统提供的下拉列表控件，单击后弹出一个对话框，显示几个供选择的选项。当选项太多时，由于手机屏幕大小有限，使用单选按钮将会占用很大的空间，而使用 Spinner 控件是一种好的解决方法。下拉列表控件需要绑定选项数据集合才能显示出选项的文本内容，系统提供了函数 fullSetList 用来绑定数据：

```
fullSetList(String id,JSONArray list)
```

函数 fullSetList 的用途是将列表数据绑定到下拉列表控件，参数 id 表示下拉列表控件的 id，参数 list 是列表项数据集合，可用数组表示。

下面是下拉列表控件的范例。

【例 18-2-16】（代码位置：\18\list 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#ff000000"
    >
  <TextView
    android:text="请选择你喜欢的城市："
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
  <Spinner
    android:id="@+id/spinner"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$mylist=array();
$mylist[0]="北京";
$mylist[1]="上海";
$mylist[2]="三亚";
$droid->fullSetList("spinner",$mylist);
$droid->eventWait(8000);
?>
```

程序运行后会提示用户选择自己喜欢的城市，用户可以单击列表，列表会弹出对话框显示“北京”、“上海”和“三亚”3 个选项，用户可以单击选项或拉动列表进行选择，运行结果如图 18-2-16 所示。

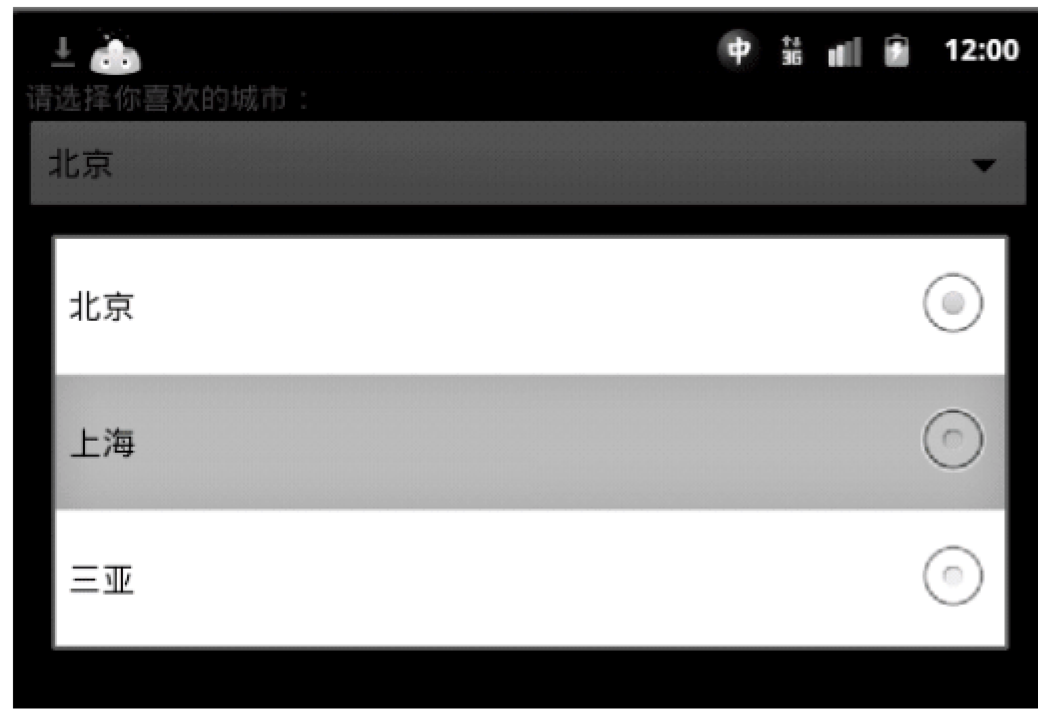


图 18-2-16 下拉列表

ListView 控件是 Android 提供的一种列表控件，经常会在应用程序中使用列表的形式

来展现一些内容供用户单击选择。它和下拉列表很相似,两者的区别是:ListView 控件会直接以列表的形式向用户显示数据,而下拉列表需要用户单击下拉列表后才以列表的形式向用户显示数据。下面是列表应用的范例,例中的属性 android:drawSelectorOnTop 用来设置选中条样式,其值可取 true 和 false,android:drawSelectorOnTop="true"表示单击某一条记录,选中条颜色会显示在最上面,记录上的文字被遮住;android:drawSelectorOnTop="false"表示单击某条记录不放,颜色会在记录的后面,成为背景色,但是记录内容的文字是可见的。例中用图像作为列表的背景。

【例 18-2-17】 (代码位置:\18\listview 目录)

文件 mylayout.xml 内容如下:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
       android:id="@+id/LinearLayout01"
       android:layout_width="fill_parent"
       android:layout_height="fill_parent"
       xmlns:android="http://schemas.android.com/apk/res/android">
    <ListView android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:id="@+id/ListView01"
              android:drawSelectorOnTop="false"
              android:background="file:///sdcard/sl4a/scripts/moon.png"
            />
</LinearLayout>
```

文件 test.php 内容如下:

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$mylist=array();
$mylist[0]="北京";
$mylist[1]="上海";
$mylist[2]="三亚";
$droid->fullSetList("ListView01",$mylist);
$droid->eventWait(8000);
?>
```

程序运行结果如图 18-2-17 所示。

18.2.11 菜单

菜单是用户界面中最常见的元素之一,使用非常频繁。系统提供了选项菜单

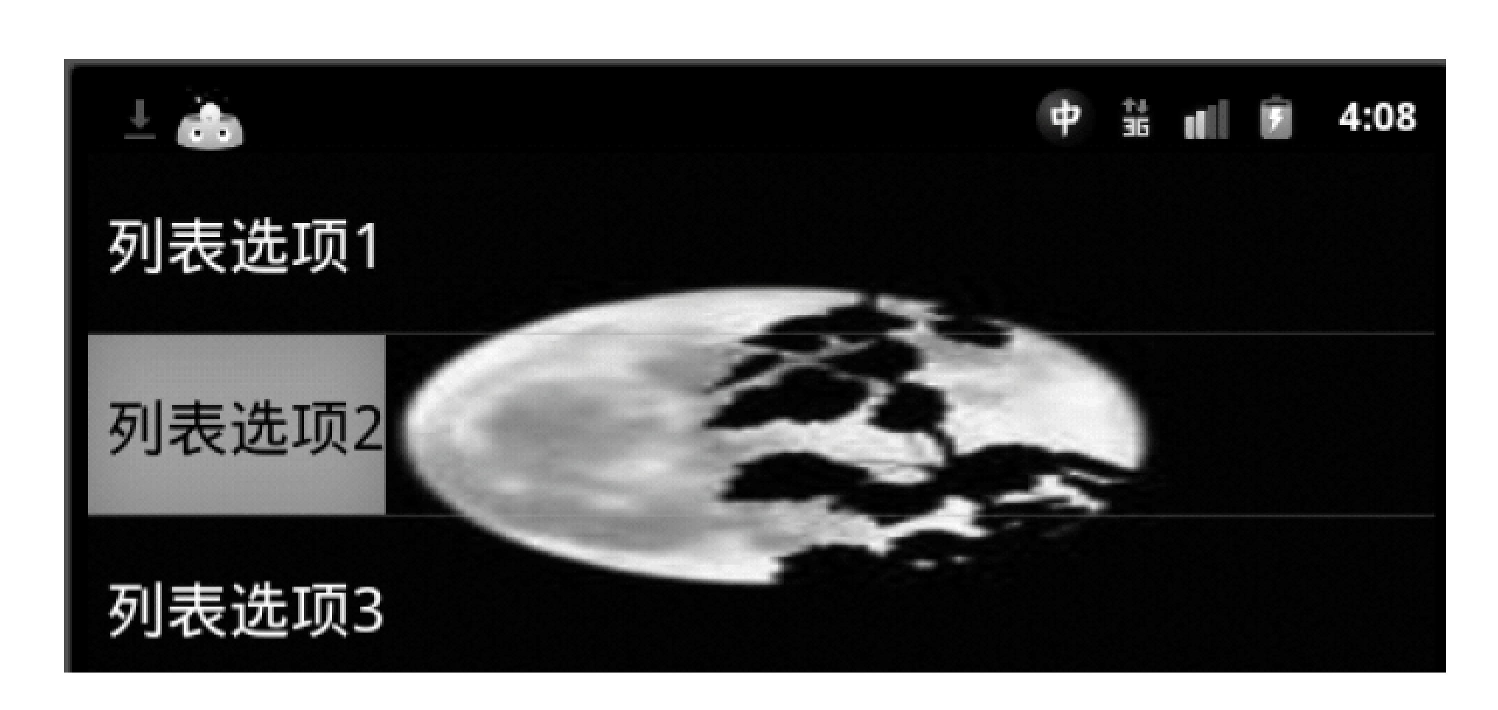


图 18-2-17 列表

(OptionsMenu)和上下文菜单(ContextMenu)。

当用户按下手机键盘上的菜单按键时,系统通常会在屏幕底部弹出一个菜单,该菜单称为选项菜单,但这个功能需要开发人员编程实现,如果在开发应用程序时没有实现该功能,那么程序运行时按下手机的菜单键是不会起作用的。选项菜单最多显示 6 个菜单项,这些菜单项可以有文字和图标;如果多于 6 项,则只会显示前面 5 个菜单项和一个扩展菜单项,第 6 项以及以后的菜单项会被隐藏,扩展菜单项会出现一个 More 文本,单击 More 会跳出一个菜单,这个菜单被称作扩展菜单(expanded menu),它包含第 6 项以及以后的菜单项。下面介绍选项菜单中的菜单项添加和删除函数。

```
addOptionsMenuItem(String label,String event,Object eventData,String iconName)
```

函数 addOptionsMenuItem 用于为选项菜单添加菜单项,参数 label 表示菜单项标签;参数 event 表示事件名,当该菜单项被单击时将产生该事件;参数 eventData 表示事件附带的数据,该参数是可选的;参数 iconName 表示图标名称,图标由系统提供,该参数是可选的,有关图标资源可参看 http://developer.android.com/reference/android/R.drawable.html。

```
clearOptionsMenu()
```

函数 clearOptionsMenu 用于删除选项菜单中的所有菜单项。

下面是选项菜单的范例。

【例 18-2-18】 (代码位置:\18\optionsmenu 目录)

文件 mylayout.xml 内容如下:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#000000"
>
  <TextView
    android:text="提示:按 MENU 按键弹出菜单。"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:textStyle="bold"
        android:textSize="20px"
    />
    <Button
        android:text="退出"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/mybutton"
        android:textStyle="bold"
        >
    </Button>
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$droid->addOptionsMenuItem("菜单项 1","item1",null,"star_on");
$droid->addOptionsMenuItem("菜单项 2","item2",null,"star_off");
$droid->addOptionsMenuItem("菜单项 3","item3",null,"ic_menu_revert");

$ret1=$droid->eventWait(100);
while($ret1['result']->data->id!='mybutton')
    $ret1=$droid->eventWait(100);
?>
```

程序运行后，单击模拟器中的 MENU 按键会弹出菜单，菜单有 3 个菜单项，每个菜单项都带有图标，要结束程序可单击“退出”按钮。其运行结果如图 18-2-18 所示。



图 18-2-18　选项菜单

上下文菜单类似于 PC 中的右键弹出菜单，上下文菜单不支持图标，SL4A 提供的上下文菜单主要用于 WebView。SL4A 内建有 HTML 解释器和 API 用于控制 WebView，从而能利用 HTML 制作程序界面，这样做的一个好处是能让 Web 开发人员快速地开发 Android 程序。下面介绍 WebView 函数和上下文菜单函数。

```
webViewShow(String url,Boolean wait)
```

函数 webViewShow 用于显示一个指定 URL 的 WebView。参数 url 表示 HTML

文件。参数 wait 是布尔类型变量，该参数用来表示是否以堵塞方式运行 WebView，值 true 表示以堵塞方式运行 WebView，值 false 表示以非堵塞方式运行，默认值是 false，该参数是可选的。

```
addContextMenuItem(String label,String event,Object eventData)
```

函数 addContextMenuItem 用于添加上下文菜单项，参数 label 表示菜单项标签，参数 event 表示事件名，单击菜单项时将产生该事件，该参数可选；参数 eventData 表示事件附带的数据，该参数可选。

```
clearContextMenu()
```

函数 clearContextMenu 用于删除上下文菜单中的所有菜单项。

下面是上下文菜单的范例。

【例 18-2-19】 （代码位置：\18\contextmenu 目录）

文件 test. htm 内容如下：

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf8"/>
    <meta name="author" content="WWW.DOWNG.COM"/>
</head>
<body>
提示：长按 2 秒弹出菜单。
</body>
</html>
```

文件 test. php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$droid->webViewShow("file:///sdcard/sl4a/scripts/test.htm");
$droid->addContextMenuItem("菜单项 1","item1",null);
$droid->addContextMenuItem("菜单项 2","item2",null);
$droid->addContextMenuItem("菜单项 3","item3",null);
$droid->eventWait(30000);
?>
```

先把文件 test. htm 复制到模拟器的 /sdcard/sl4a/scripts/目录下，然后再运行程序 test. php。程序运行后会显示 test. htm 界面，如果用户按住屏幕 2 秒则会弹出菜单。其运行结果如图 18-2-19 所示。



图 18-2-19　上下文菜单

18.3 事件处理

18.3.1 事件API

现代的用户界面,都是以事件驱动的方式来实现人机交互的,而 Android 上的一套 UI 组件,无非就是投递鼠标和键盘等事件,每个组件在收到事件之后做相应的处理。PFA 提供基于监听事件的处理模型,在监听器模型中,主要涉及 3 类对象,具体如下。

(1) 事件源:产生事件的来源,通常是各种组件,如按钮和窗口等,这些事件通常是系统产生的。此外,用户可以自定义事件并投递事件从而驱动程序运行。

(2) 事件:事件由事件名和事件具体信息两部分组成,它们对事件进行了较详细的描述,事件是事件源和事件监听器的中介。

(3) 事件监听器:负责监听事件源发生的事件,并对不同的事件做相应的处理。

基于监听事件的处理机制:事件源负责产生事件或接收系统级事件并投递事件到事件队列中,事件监听器负责监听事件,如果监听到事件,则对事件进行响应和处理。这种处理方式将事件源和事件监听器分离,有利于提供程序的可维护性。

系统提供了许多函数用来管理事件,这些函数的功能主要是向事件队列中投递事件以及从事件队列中删除事件和取出事件,下面介绍事件函数。

```
eventClearBuffer()
```

函数 eventClearBuffer 用于清除事件缓冲区中的所有事件。

```
eventPoll(Integer number_of_events)
```

函数 eventPoll 用于从事件缓冲区中获取投递时间最久的若干事件,事件将从事件缓冲区中删除,事件数量由参数 number_of_events 指定,number_of_events 默认值为 1,该参数是可选的,函数以键值列表方式返回事件。

```
eventPost(String name,String data,Boolean enqueue)
```

函数 eventPost 用于将一个事件放投递到事件队列中,参数 name 表示事件名,参数 data 表示事件数据,参数 enqueue 是布尔值,其值为 false 表明只分发事件而不把事件投递到事件队列中,其值为 true 表示事件要投递到事件队列中,默认值是 false,该参数是可选的。

```
eventWait(Integer timeout)
```

函数 eventWait 用于等待事件,参数 timeout 表示最大等待时间,如果在指定等待时间内没有事件,则函数将以堵塞方式运行直到等待时间结束;如果在指定等待时间范围内有事件,则返回事件并从缓冲区中删除该事件;如果未指定 timeout 则一直等待直到有事件到来;如果 timeout 值为 0 则表示不等待。等待时间的单位是毫秒,该参数是可选的。如果有事件产生,该函数将返回事件,事件包含事件名称、事件附加数据和事件产生

时间，分别用name、data和time属性表示，不同事件的名称、附加数据和时间都不同，特别是data属性，它可以值的形式和以对象的形式来封装数据，当以对象的形式封闭事件时，对不同名称的事件，data对象中的属性名不同。

下面是事件处理的范例，该程序模拟了文件下载并显示文件下载进度，它同时扮演了事件源和事件监听器两个角色。

【例18-3-1】 （代码位置：\18\event\test. php）

```
<?php
require_once("Android.php");
$droid=new Android();
$droid->dialogCreateHorizontalProgress("文件下载","正在下载,请等待...");
$droid->dialogShow();
$items=rand(0,10);                         //模拟文件下载和计算文件下载进度
$droid->eventPost("DownloadProgress","$items",true);
$flag=true;
while($flag)
{
    $ret=$droid->eventWait();              //等待事件
    switch($ret['result']->name)
    {
        case "DownloadFinish":             //下载结束事件处理
            $flag=false;
            break;
        case "DownloadProgress":           //正在下载事件处理
            $items=$ret['result']->data;
            $droid->dialogSetCurrentProgress($items);
            break;
        default: $flag=false;break;        //其他事件处理
    }

    $items+=rand(0,10);                    //模拟文件下载并累计下载进度
    if($items>100)
        $droid->eventPost("DownloadFinish","",true);
    else $droid->eventPost("DownloadProgress","$items",true);
}
$droid->dialogDismiss();
?>
```

程序运行后会模拟文件下载并计算下载进度，同时进度条也会实时地指示下载进度。如果把程序中的true改为false，那么会发现进度条并不会前进，因为这时候函数eventPost并没有把事件投递到事件队列中，函数eventWait从事件队列中将读不到事件。

18.3.2 菜单事件处理

当用户单击菜单项时，系统会产生一个菜单项单击事件，程序可通过函数获取事件并进行处理。下面是菜单事件处理的范例，该例具有通过菜单对文本进行复制、粘贴和设置字体颜色等功能。

【例 18-3-2】 （代码位置：\18\menuevent 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#000000"
>
  <TextView
    android:text="你可在编辑框里输入文本,点击菜单按键进行复制、粘贴和设置颜色等操作。"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
  />
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:id="@+id/editText1"
    android:text="你可在这里输入文本!"
  />
  <TextView
    android:text="这是 TextView 范例,本范例的文字居中对齐。"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#80ff0000"
    android:background="#80ff00"
    android:gravity="center"
    android:id="@+id/textView1"
  />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
```

```
$droid->addOptionsMenuItem("复制","copy",null,"star_on");
$droid->addOptionsMenuItem("粘贴","paste",null,"star_on");
$droid->addOptionsMenuItem("设置红色","setColor","#80ff0000","star_on");
$droid->addOptionsMenuItem("设置蓝色","setColor","#800000ff","star_on");
$droid->addOptionsMenuItem("退出","exit",null,"ic_menu_revert");
$ret=$droid->eventWait(10000);
while($ret['result']->name!='exit')
{
    switch($ret['result']->name)
    {
        case 'copy':
                    $text=$droid->fullQueryDetail("editText1");
                    $text=$text['result']->text;
                    break;
        case 'paste':
                    $droid->fullSetProperty("textView1","text",$text);
                    break;
        case 'setColor':
                    $color=$ret['result']->data;
                    $droid->fullSetProperty("textView1","textColor",$color);
                    break;

    }
    $ret=$droid->eventWait(1000);
}
?>
```

程序运行后，用户可先在文本编辑框中输入文本，然后单击菜单按键弹出菜单，最后单击菜单项进行各种操作，其运行结果如图 18-3-1 所示。



图 18-3-1　菜单事件处理

18.3.3　按钮事件处理

当按钮被单击时，系统将产生单击事件，事件 Name 统一为 click，事件附带的数据

Data 的格式会因为按钮的不同类型而有所不同,具体如表 18-3-1 至表 18-3-6 所示。

表 18-3-1 Button 类型的 Data 格式

属 性	id	type	visibility	text
含义和值	按钮标识符	Button	0 或 1	按钮文本

表 18-3-2 ImageButton 类型的 Data 格式

属 性	id	type	visibility
含 义	标识符	ImageButton	0 或 1

表 18-3-3 RadioButton 类型的 Data 格式

属 性	id	type	visibility	text	checked
含 义	标识符	RadioButton	0 或 1	按钮文本	"true"或"false"

表 18-3-4 CheckBox 类型的 Data 格式

属 性	id	type	visibility	text	checked
含 义	标识符	CheckBox	0 或 1	按钮文本	"true"或"false"

表 18-3-5 ToggleButton 类型的 Data 格式

属 性	id	type	visibility	text	checked
含 义	标识符	ToggleButton	0 或 1	按钮文本	"true"或"false"

表 18-3-6 DatePicker 和 TimePicker 类型的 Data 格式

属 性	id	type	visibility
含 义	标识符	NumberPickerButton	0 或 1

注意: TimePicker 控件有一个"上午"或"下午"按钮,当单击该按钮时,返回的事件和普通按钮产生的事件一样,不同的是,按钮文本是"上午"或"下午"。

下面是按钮事件处理的范例,例中有三个 Button 和一个 CheckBox,单击不同的按钮系统都会产生不同的事件,程序会对事件进行判断并进行相应的处理,第一个 Button 用于在文本框中显示内容,第二个 Button 用于在编辑框中输入内容,第三个 Button 用于退出程序,CheckBox 用于把编辑框中的内容显示到文本框中。

【例 18-3-3】 (代码位置:\18\buttonevent\test.php)

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=<<<XMLCONTENT
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/background"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ff000000">
    <LinearLayout android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/linearLayout1">
            <Button android:id="@+id/button1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="在文本框中显示内容">
            </Button>
            <Button android:id="@+id/button2"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="在编辑框中显示内容">
            </Button>
            <Button android:id="@+id/button3"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="退出">
            </Button>
    </LinearLayout>
    <TextView android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:id="@+id/textView1"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:gravity="center_vertical|center_horizontal|center">
    </TextView>
    <EditText android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/editText1"
            android:tag="Tag Me"
            android:inputType="textCapWords|textPhonetic|number">
            <requestFocus></requestFocus>
    </EditText>
    <CheckBox android:layout_height="wrap_content"
        android:id="@+id/checkBox1"
        android:layout_width="234dp"
        android:text="显示编辑框文本(先在编辑框中输入内容)"
        android:checked="true">
```

```
    </CheckBox>
</LinearLayout>
XMLCONTENT;
$droid->fullShow($layout);
while(true)
{
    $event=$droid->eventWait();
    $event=$event['result'];

    if($event->name=="click")
      $id=$event->data->id;
    if($id=="button3")
        return;
    if($id=="button2")
        $droid->fullSetProperty("editText1","text","OK has been pressed");
    if($id=="button1")
        $droid->fullSetProperty("textView1","text","Other stuff here");
    if($id=="checkBox1")
    {
        $ret=$droid->fullQueryDetail("editText1");
        $droid->fullSetProperty("textView1","text","Data entered="
        .$ret['result']->text);
    }
}
?>
```

程序运行结果如图18-3-2所示，用户单击第一个Button、第二个Button和CheckBox按钮会在文本框和编辑框中产生不同的内容，单击第三个Button会结束程序。



图 18-3-2　按钮事件处理

18.3.4　列表事件处理

ListView选项被单击后会产生itemclick事件，该事件附加的数据格式如表18-3-7所示。

表 18-3-7　ListView 的 Data 格式

属性	id	type	visibility	position	selectedItemPosition
含义	标识符	ListView	0	选中项位置	

下面是列表事件处理的范例，例中先弹出列表，然后用户单击列表选项，最后程序显示用户的选项。

【例 18-3-4】（代码位置：\18\listevent 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
  <ListView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ListView01"
            android:drawSelectorOnTop="false"
            android:background="file:///sdcard/sl4a/scripts/moon.png"
            />

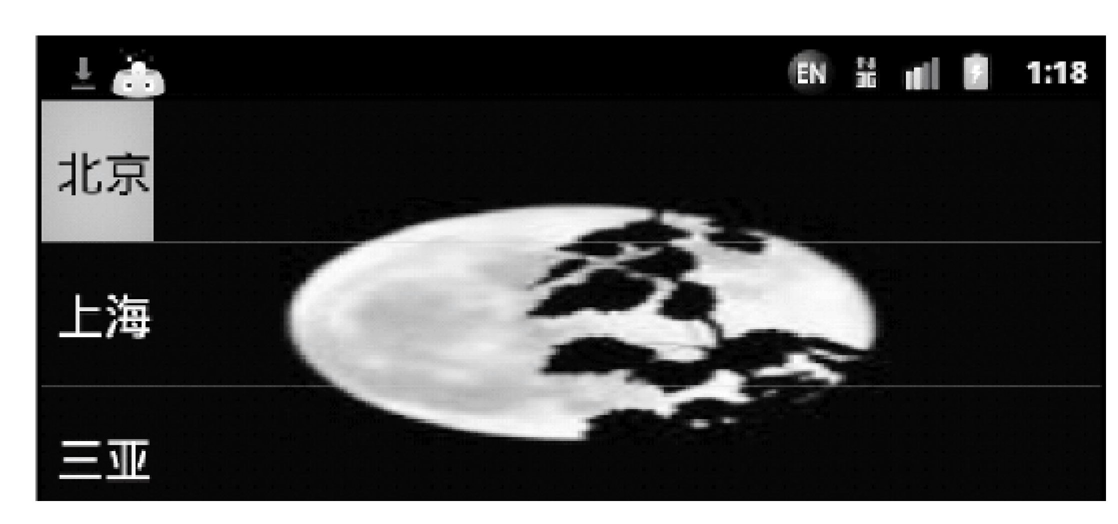
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$mylist=array();
$mylist[0]="北京";
$mylist[1]="上海";
$mylist[2]="三亚";
$droid->fullSetList("ListView01",$mylist);
$event=$droid->eventWait(8000);
$droid->fullDismiss();
$item=$event['result']->data;
$item=$item->position;
$droid->dialogCreateAlert("选项提示","您点击了选项$item");
$droid->dialogSetPositiveButtonText("确定");
$droid->dialogSetNegativeButtonText("退出");
```

```
$droid->dialogShow();
$droid->dialogGetResponse();
?>
```

程序运行结果如图 18-3-3 所示，图 18-3-3(a)是列表，用户选择了“北京”选项，图 18-3-3(b)显示了用户选项。



(a) 列表选项



(b) 选项提示

图 18-3-3 列表事件处理

18.3.5 键盘事件处理

下面是键盘事件处理的范例，用户可以关闭或打开声音按键默认行为，从而禁止或允许声音按键调整声音。

【例 18-3-5】 （代码位置：\18\keyevent 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#000000"
>
  <TextView
    android:text="请按模拟器键盘上的声音按键观察是否会跳出声音调整窗口。"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
  />
  <ToggleButton
    android:id="@+id/mytogglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="覆盖声音按键"
    android:textOff="恢复声音按键"
    android:checked="true">
  </ToggleButton>
  <Button
```

```
        android:text="退出"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/mybutton"
    >
    </Button>
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);

$ret=$droid->eventWait(10000);
$event=$ret['result'];
while($event->data->id!='mybutton')
{
    if(($event->name=='click')&&($event->data->id=='mytogglebutton'))
    {
        $mytogglebutton=$droid->fullQueryDetail("mytogglebutton");
        $mytogglebutton=$mytogglebutton['result'];
        if($mytogglebutton->checked=='true')
        {
            $keys=array();
            $keys[0]=24;
            $keys[1]=25;
            $keys=$droid->fullKeyOverride($keys,false);
        }
        else
        {
            $keys=array();
            $keys[0]=24;
            $keys[1]=25;
            $keys=$droid->fullKeyOverride($keys,true);
        }
    }
    $ret=$droid->eventWait(10000);
    $event=$ret['result'];
}
?>
```

程序运行后,可先单击"覆盖声音按键"或"恢复声音按键"按键,再单击模拟器上的声音按键,观察系统是否跳出声音调整窗口,会发现在覆盖声音状态下不会跳出声音调整窗口,而在恢复声音状态会跳出声音调整窗口。程序运行结果如图 18-3-4 所示。

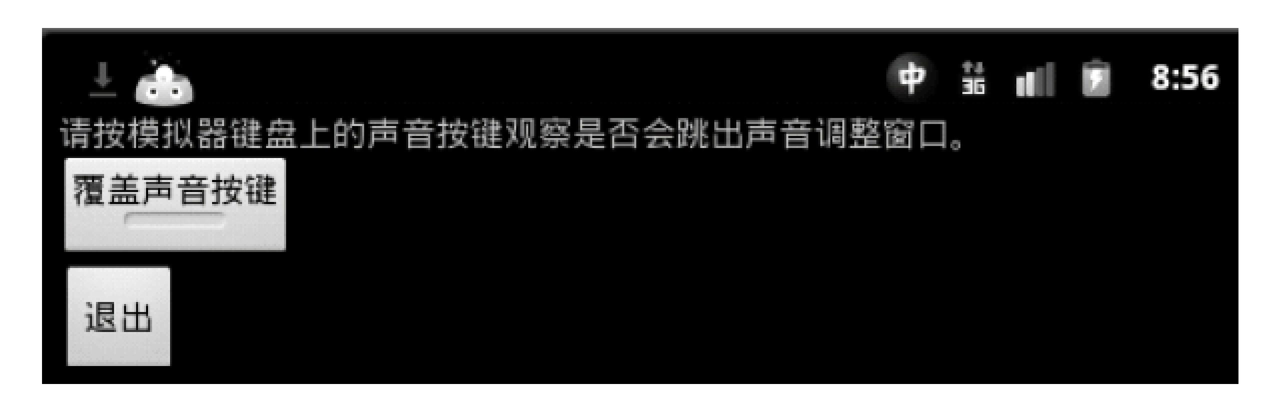


图 18-3-4 键盘事件处理

第 19 章 chapter 19

通信、多媒体和语音处理

19.1 电话通信

19.1.1 手机基本概念

IMEI(International Mobile Equipment Identity)是国际移动设备身份码的缩写,俗称"串号",它存储在手机的 EEPROM 中。IMEI 与每台手机一一对应,而且该码是全世界唯一的,IMEI 码由 GSM(全球移动通信协会)统一分配,这个号码从生产到交付使用都将被制造商所记录。IMEI 由 15 位数字组成,前 6 位数是型号核准号码(Type Approval Code,TAC),一般代表机型;接着的 2 位数是最后装配号(Final Assembly Code,FAC),一般代表产地;之后的 6 位数是串号(SNR),一般代表生产顺序号;最后 1 位数(SP)通常是 0,为检验码,目前暂备用。

MEID(Mobile Equipment IDentifier)是全球唯一的 56 位移动终端标识号,标识号会被烧入终端里,并且不能被修改,可用来对移动式设备进行身份识别和跟踪,MEID 主要分配给 CDMA 制式的手机。

IMSI(International Mobile Subscriber Identification)Number 中文翻译为国际移动用户识别码,是区别移动用户的标志,储存在 SIM 卡中,可用于区别移动用户的有效信息。IMSI 码共有 15 位,其结构是 MCC+MNC+MSIN。MCC(Mobile Country Code)中文翻译为移动国家码,MCC 的资源由国际电联(ITU)统一分配和管理,唯一识别移动用户所属的国家,共 3 位,中国为 460;MNC(Mobile Network Code)中文翻译为移动网络码,共 2 位,中国移动 TD 系统使用 00,中国联通 GSM 系统使用 01,中国移动 GSM 系统使用 02,中国电信 CDMA 系统使用 03。MSIN(Mobile Subscriber Identification Number)是移动用户识别号码,共 10 位,用来标识一个移动网络中的移动用户身份。一个典型的 IMSI 号码为 460030912121001。

MSISDN(Mobile Subscriber International ISDN/PSTN number)是主叫用户呼叫 GSM PLMN 中的一个移动用户所需拨的号码,它由 CC+NDC+SN 组成。CC(Country Code)是国家码,例如,中国的国家码为 86。NDC(National Destination Code)是国内接入号,如中国的 NDC 目前有 139、138、137、136 和 135 等。SN(Subscriber Number)是客户号码,在中国,该码长度为 8。NDC+SN 就是通常所说的手机号,例如,139××××××××。

IMSI 和 MSISDN 的作用是相同的，它们都是用来识别移动用户的，不同的是，MSISDN 用手机号识别移动用户，IMSI 从物理上识别移动用户，即移动网络运营商使用 IMSI 识别移动用户，而普通人使用 MSISDN 来识别移动用户。

19.1.2 手机 API

如果客户管理等应用程序能提供拨打电话、来电显示、查询移动网络、手机设备和 SIM 卡信息等功能，相信应用将会较好地吸引用户的目光。开发人员可以借助手机 API 轻易地实现这些功能。

```
checkNetworkRoaming()
```

checkNetworkRoaming 函数检测手机设备在网络中是否处于漫游状态，如果手机当前的 GSM 网络连接处于漫游状态，则返回 true。该函数需要在真实设备下运行才有效。

```
getCellLocation()
```

getCellLocation 函数返回当前移动终端的位置。该函数需要在真实设备下运行才有效。

```
getDeviceId()
```

getDeviceId 函数返回当前移动终端的唯一标识，例如，GSM 网络返回 IMEI 码，CDMA 网络返回 MEID 码。当移动终端的标识不可获得时，返回 null。该函数需要在真实设备下运行才有效，如果在模拟器下运行将返回 15 个 0。

```
getDeviceSoftwareVersion()
```

getDeviceSoftwareVersion 函数返回移动终端的软件版本，例如，GSM 手机的 IMEI/SV 码。如果软件版本不可获得，则返回 null。该函数需要在真实设备下运行才有效。

```
getLine1Number()
```

getLine1Number 函数在一行中返回手机号码，例如返回 GSM 的 MSISDN 号。如果获取不到则返回 null。

```
getNeighboringCellInfo()
```

getNeighboringCellInfo 函数返回当前移动终端附近移动终端的信息。

```
getNetworkOperator()
```

getNetworkOperator 函数返回当前已注册网络运营商的数值名称（MCC＋MNC），该数值由移动国家码＋移动网络码构成。

```
getNetworkOperatorName()
```

getNetworkOperatorName 函数返回当前注册网络运营商的字母名称。

```
getNetworkType()
```

getNetworkType 函数返回移动终端当前使用的无线技术(网络类型)。

```
getPhoneType()
```

getPhoneType 函数返回移动终端的类型。

```
getSimCountryIso()
```

getSimCountryIso 函数返回符合 ISO 标注的国家码,等同于 SIM 卡提供商的国家码。

```
getSimOperator()
```

getSimOperator 函数返回 SIM 卡的 MCC+MNC 组合,即 SIM 提供商的移动国家码和移动网络码的组合,这个组合是一个 5 位或 6 位的数字。

```
getSimOperatorName()
```

getSimOperatorName 函数返回 SIM 卡的服务运营商的名称(SPN)。

```
getSimSerialNumber()
```

getSimSerialNumber 函数返回 SIM 卡的序列号,如果无法获得则返回 null。

```
getSimState()
```

getSimState 函数返回 SIM 卡状态。

```
getSubscriberId()
```

getSubscriberId 函数返回用户唯一标识,例如 GSM 网络的 IMSI 编号。如果不可获得则返回 null。

```
getVoiceMailAlphaTag()
```

getVoiceMailAlphaTag 函数获取语音信箱号码关联的字母标识。

```
phoneCall(String uri)
```

phoneCall 函数通过 URI 调用一个联系人或电话号码。

```
phoneCallNumber(String phonenumber)
```

phoneCallNumber 函数拨打一个电话号码。

```
phoneDial(sring uri)
```

phoneDial 函数通过 URI 调用一个联系人或拨号。

```
phoneDialNumber(String phone number)
```

phoneDialNumber 函数拨打一个电话号码。

```
readPhoneState()
```

readPhoneState 函数返回当前移动终端状态和呼叫号码，分别存储在 state 和 incomingNumber 属性中。

```
startTrackingPhoneState()
```

startTrackingPhoneState 函数开始追踪移动终端状态。

```
stopTrackingPhoneState()
```

stopTrackingPhoneState 函数停止追踪移动终端状态。

下面是使用手机 API 读取 SIM 号等信息的范例，范例中的 API 可以运行在模拟器和真实手机上，范例可读出模拟器和手机设备的 SIM 号和网络运营商等内容。范例由文件 mylayout.xml 和 test.php 组成，前者用于描述主界面，后者用于显示 SIM 号和网络运营商等。

【例 19-1-1】 (代码位置：\19\phone 目录)

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="#ff000000"
  >
  <ListView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ListView01"
            android:drawSelectorOnTop="false"
  />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$ret=$droid->fullShow($layout);
$mylist=array();
$mylist[0]="读取手机号码";
$mylist[1]="读取移动网络运营商号";
$mylist[2]="读取移动网络运营商名称";
$mylist[3]="读取移动网络类型";
$mylist[4]="读取移动终端类型";
$mylist[5]="读取国家码";
```

```
$mylist[6]="读取 SIM 号";
$mylist[7]="读取 SIM 卡运营商";
$mylist[8]="读取 SIM 卡序列号";
$mylist[9]="读取 SIM 状态";
$mylist[10]="读 IMSI 号";
$droid->fullSetList("ListView01",$mylist);
$event=$droid->eventWait();
$droid->fullDismiss();
$item=$event['result']->data;
$item=$item->position;
switch($item)
{
    case 0:
            $ret=$droid->getLine1Number();
            $droid->dialogCreateAlert("测试函数 getLine1Number","本机号码：$ret
                [result]");
            break;
    case 1:
            $ret=$droid->getNetworkOperator();
            $droid->dialogCreateAlert("测试函数 getNetworkOperator","移动网络运
                营商号：$ret[result]");
            break;
    case 2:
            $ret=$droid->getNetworkOperatorName();
            $droid->dialogCreateAlert("测试函数 getNetworkOperatorName","移动
                网络运营商名：$ret[result]");
            break;
    case 3:
            $ret=$droid->getNetworkType();
            $droid->dialogCreateAlert("测试函数 getNetworkType","网络类型：$ret
                [result]");
            break;
    case 4:
            $ret=$droid->getPhoneType();
            $droid->dialogCreateAlert("测试函数 getPhoneType","移动终端类型：
                $ret[result]");
            break;
    case 5:
            $ret=$droid->getSimCountryIso();
            $droid->dialogCreateAlert("测试函数 getSimCountryIso","国家码：$ret
                [result]");
            break;
    case 6:
```

```
            $ret=$droid->getSimOperator();
            $droid->dialogCreateAlert("测试函数 getSimOperator","SIM号:$ret
                [result]");
            break;
        case 7:
            $ret=$droid->getSimOperatorName();
            $droid->dialogCreateAlert("测试函数 getSimOperatorName","SIM卡运营
                商:$ret[result]");
            break;
        case 8:
            $ret=$droid->getSimSerialNumber();
            $droid->dialogCreateAlert("测试函数 getSimSerialNumber","SIM卡序列
                号:$ret[result]");
            break;
        case 9:
            $ret=$droid->getSimState();
            $droid->dialogCreateAlert("测试函数 getSimState","SIM状态:$ret
                [result]");
            break;
        case 10:
            $ret=$droid->getSubscriberId();
            $droid->dialogCreateAlert("测试函数 getSubscriberId","IMSI号:$ret
                [result]");
            break;
        default:
            break;
    }
    $droid->dialogSetPositiveButtonText("确定");
    $droid->dialogSetNegativeButtonText("退出");
    $droid->dialogShow();
    $droid->dialogGetResponse();
    ?>
```

程序运行后会跳出一个列表,可单击列表项对手机 API 进行测试,单击后系统会跳出对话框并给出 API 的返回结果。程序在模拟器上的运行结果如图 19-1-1 所示,图 19-1-1(a)是功能列表,图 19-1-1(b)是单击“读出手机号码”选项后显示的模拟器手机号。注意,运行该范例前需要把文件 mylayout.xml 和 test.php 复制到模拟器或手机的/sdcard/sl4a/scripts/目录,下面章节的范例做法相同。

19.1.3 监听手机状态实例

下面的程序可以对手机状态进行监听,程序运行后并不会显示软件界面,只有当手机状态发生变化时才会以对话框方式显示新状态。程序使用函数 readPhoneState 读取手机状



(a) 功能列表



(b) 手机号

图 19-1-1 手机 API 测试

态，在读手机状态前后应该调用函数 startTrackingPhoneState 和 stopTrackingPhoneState 进入监听跟踪和退出监听跟踪状态。

【例 19-1-2】 （代码位置：\19\testTrackPhoneState.php）

```
<?php
require_once("Android.php");
$droid=new Android();
$lastState='init';                        //初始状态
$droid->startTrackingPhoneState();
$droid->addOptionsMenuItem("退出","exit",null,"ic_menu_revert");
$ret=$droid->eventWait(10000);
while($ret['result']->name!='exit')
{
    $ret=$droid->readPhoneState();
    $ret=$ret['result'];

    $state=$ret->state;
    $incomingNumber=$ret->incomingNumber;
    //当前状态和最近状态不一致则显示最新状态
    if($lastState !=$state)
    {
        $droid->dialogCreateAlert("电话状态检测","电话状态：$state \n 来电号码：
        $incomingNumber");
        $droid->dialogSetPositiveButtonText("确定");
        $droid->dialogSetNegativeButtonText("退出");
        $droid->dialogShow();
        $ret=$droid->dialogGetResponse();
        if($ret['result']->which=='negative')
            break;
    }
    $lastState=$state;
    $ret=$droid->eventWait(1000);
```

```
}
$droid->stopTrackingPhoneState();
?>
```

程序运行后并不会显示界面，但可以在状态栏上看到 SL4A 图标，该图标表明该程序仍然在运行当中。程序运行后，可以用两个模拟器拨打电话，操作方法可参看前面章节内容。当来电显示、呼叫电话、挂断电话和通话时，会看到有对话框跳出来提示手机状态，如果单击“确定”按钮则继续监听电话状态，单击“退出”按钮将结束程序运行。当手机来电时，程序会显示 ringing 状态和来电号码，其运行结果如图 19-1-2 所示。

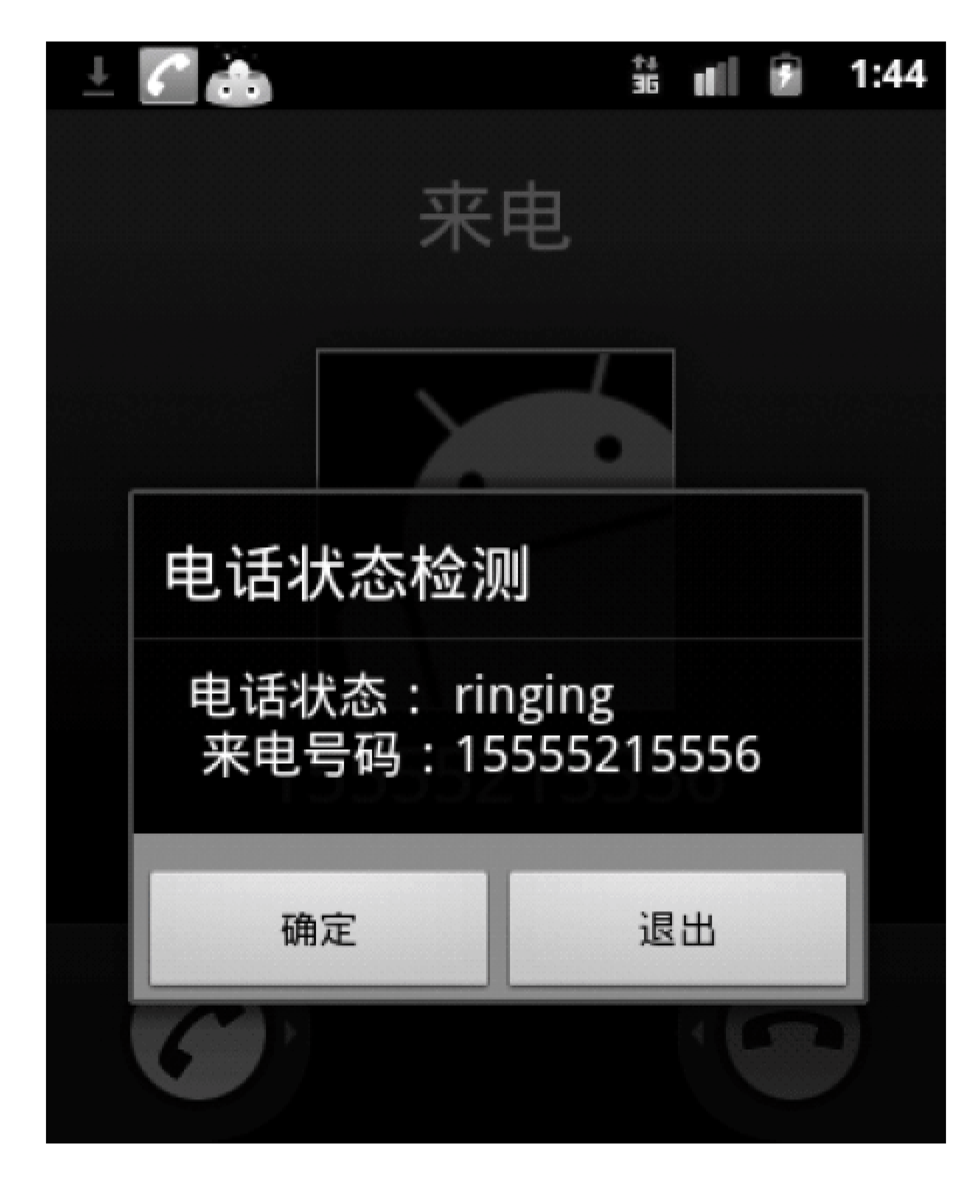


图 19-1-2 来电状态

19.1.4 拨打电话实例

下面是拨打电话的范例，例中有一个文本编辑框和一个电话按钮，用户可先在文本编辑框内输入电话号码，再单击“拨打电话”按钮拨打电话。

【例 19-1-3】 （代码位置：\19\CallPhone）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:background="#ff000000" android:gravity="center">
  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="text"
    android:id="@+id/phonneNumber"
    android:text="5556"
    />
  <Button
    android:text="拨打电话"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/dial"
    android:textStyle="bold"
    android:textSize="20px"
    >
  </Button>
```

```
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$event=$droid->eventWait();
$event=$event['result'];
if(($event->name=="click")&&($event->data->id=="dial"))
{
    $phonneNumber=$droid->fullQueryDetail("phonneNumber");
    $phonneNumber=$phonneNumber['result']->text;
    $droid->phoneCallNumber($phonneNumber);
}
?>
```

首先需要把程序复制到模拟器 A 上，然后再运行另一个模拟器 B，模拟器 B 的号码是 5556，最后运行程序。这种方法需要反复开启和关闭模拟器，如果觉得这种测试方法麻烦，可以先同时运行两个模拟器，然后把程序复制到模拟器 A 上，关于如何在开启两个模拟器的情况下复制程序到指定模拟器，请参阅前面章节的内容，最后再运行程序。程序在模拟器上的运行结果如图 19-1-3 所示，先在编辑框中输入电话号码 5556，然后单击“拨打电话”按钮，应用将直接拨打手机号码为 5556 的模拟器。



图 19-1-3　拨打电话

19.2　短信通信

19.2.1　短消息 API

通常，用户除发送短消息外，还会使用标识短消息为已阅读状态、查看短消息数量和查看短消息服务中心等功能。要实现这些功能，开发人员需要掌握 Android 短消息格式和短消息 API 等内容，下面介绍这方面的内容。

```
smsDeleteMessage(Integer id)
```

smsDeleteMessage 函数的功能是删除指定的短信息，如果信息被成功删除则返回 true。参数 id 是短信息标识符。

```
smsGetAttributes()
```

smsGetAttributes 函数的功能是返回所有可能的短信息属性列表。

Android 短消息的格式是固定的，短消息字段比较多，但不是每个字段都是必填的。当系统收发短消息时，会把短消息属性值保存在数据库中，该数据库文件是/data/data/com. android. providers. telephony/mmssms. db。虽然不能直接访问数据库，但可以通过函数 smsGetAttributes 查看短消息格式，读出短消息所有的属性。短消息的属性如表 19-2-1 所示。

表 19-2-1　短消息属性

属性名	含　义
_id	短消息序号，例如，100
thread_id	对话的序号，例如，100
address	发件人地址，即手机号，例如，139××××××××
person	发件人，若返回一个数字，就是联系人列表里的序号，若返回 null 则是陌生人
date	日期，long 型，如 1256539465022
protocol	协议，值为 0 表示 SMS_RPOTO，为 1 表示 MMS_PROTO
read	是否已阅读，值为 0 表示未读，为 1 表示已读
status	状态，－1 表示已接收，0 表示完成，64 表示未定，128 表示失败
type	消息类型，1 表示接收到的消息，2 表示已发出的消息
subject	主题
body	短消息内容
service_center	短信服务中心号码，例如，＋8613800755500
locked	是否锁住短消息标记
error_code	错误代码

```
smsGetMessageById(Integer id,JSONArray attributes)
```

smsGetMessageById 函数的功能是返回指定短信息的属性。参数 id 表示短消息标识符；参数 attributes 表示属性列表，该参数可选，可以通过设置该参数读出短消息的指定属性值。

```
smsGetMessageCount(Boolean unreadOnly,String folder)
```

smsGetMessageCount 函数的功能是返回短信息的数目。参数 unreadOnly 是布尔值，表示是否读取已阅读的短信息，值 false 表示未阅读，值 true 表示已阅读；参数 folder 表示信息存储目录，该参数可选，默认值是 inbox。

```
smsGetMessageIds(Boolean unreadOnly,String folder)
```

smsGetMessageIds 函数的功能是返回所有短信息的 id。参数 unreadOnly 是布尔值，表示是否读取已阅读的短信息，值 false 表示未阅读，值 true 表示已阅读；参数 folder 表示短消息目录，该参数可选，默认值是 inbox。Android 提供的短消息目录有 sent、draft、outbox、failed 和 queued，分别表示已发送、草稿、发件箱、发送失败和待发送列表目录。

```
smsGetMessages(Boolean unreadOnly,String folder,JSONArray attributes)
```

smsGetMessages 函数的功能是返回所有短信息的列表。参数 unreadOnly 是布尔值，表示是否读取已阅读的短信息，值 false 表示未阅读，值 true 表示已阅读；参数 folder 表示信息存储目录，该参数可选，默认值是 inbox。

```
smsMarkMessageRead(JSONArray ids,Boolean read)
```

smsMarkMessageRead 函数的功能是将短信息标记为已读。参数 ids 表示短消息标识符列表；参数 read 是布尔值，表示是标识短消息为已读或未读，其返回值是被标记为已读的短消息的序号。

```
smsSend(String destinationAddress,String text)
```

smsSend 函数的功能是发送一条短信。参数 destinationAddress 表示收件人地址，通常是电话号码；参数 text 是短信内容。需要注意的是，要控制短消息长度和获取短消息的发送状态。

19.2.2 短消息范例

下面是读取短消息标识符的范例。

【例 19-2-1】 （代码位置：\19\ReadMsg 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="match_parent"
  android:layout_height="match_parent" android:background="#ff000000">
    <TextView
        android:text="读取短消息标识符范例"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25px"
        android:id="@+id/TextView01"
    />
    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/ListView01"
        android:drawSelectorOnTop="false"
    />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
    require_once("Android.php");
```

```
    $droid=new Android();
    $ids=$droid->smsGetMessageIds(false);
    $ids=$ids['result'];
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$ids);
    $droid->eventClearBuffer();
    $event=$droid->eventWait();
?>
```

程序运行后会以列表方式显示所有短消息的标识符，运行结果如图 19-2-1 所示，图中的数字 3、2 和 1 是短消息标识符。



图 19-2-1 短消息标识符

下面是另一个短消息的范例，范例实现了读取短消息总数量、读取短消息所有属性、读取所有短消息内容以及删除短消息等功能。

【例 19-2-2】 （代码位置：\19\ReadMsgAttribute 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="#ff000000"
  >
  <ListView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ListView01"
            android:drawSelectorOnTop="false"
          />
</LinearLayout>
```

文件 test. php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$mylist=array();
$mylist[0]="读取短消息数量";
$mylist[1]="读取短消息属性";
$mylist[2]="读取所有短消息内容";
```

```
$mylist[3]="删除短消息";
$droid->fullSetList("ListView01",$mylist);
$event=$droid->eventWait();
$item=$event['result']->data;
$item=$item->position;

switch($item)
{
    case 0:
            GetMessageCount($droid);
            break;
    case 1:
            showMessageAttrs($droid);
            break;
    case 2:
            showAllMessage($droid);
            break;
    case 3:
            delMessage($droid);
            break;
    default:
            break;
}
function GetMessageCount($droid)
{
    $ret=$droid->smsGetMessageCount(false);
    $count=$ret['result'];
    $droid->dialogCreateAlert("短消息数量"," $count ");
    $droid->dialogSetPositiveButtonText("确定");
    $droid->dialogSetNegativeButtonText("退出");
    $droid->dialogShow();
    $droid->dialogGetResponse();
}
function showMessageAttrs($droid)
{
    $ret=$droid->smsGetAttributes();
    $ret=$ret['result'];

    $attrs=array();
    foreach($ret as $key=>$value)
    {
        $attrs[]=$value;
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
```

```
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$attrs);
    $droid->eventClearBuffer();
    $droid->eventWait();
}
function showAllMessage($droid)
{
    $ret=$droid->smsGetMessageIds(false);
    $ret=$ret['result'];
    $sms=array();
    foreach($ret as $key=>$value)
    {
    $tmp=$droid->smsGetMessageById($value);
    $body=$tmp['result']->body;
    $address=$tmp['result']->address;
    $sms[]="短消息内容：$body \n 发信号码：$address";
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$sms);
    $droid->eventClearBuffer();
    $droid->eventWait();
}
function delMessage($droid)
{
    $ret=$droid->smsGetMessageIds(false);
    $ret=$ret['result'];
    $sms=array();
    foreach($ret as $key=>$value)
    {
        $tmp=$droid->smsGetMessageById($value);
        $body=$tmp['result']->body;
        $address=$tmp['result']->address;
        $sms[]="短消息内容：$body \n 发信号码：$address";
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$sms);
    $droid->eventClearBuffer();
    $event=$droid->eventWait();
    while($event['result']->name !='itemclick')
        $event=$droid->eventWait();
    $data=$event['result']->data;
    $position=$data->position;
```

```
    $droid->smsDeleteMessage($ret [$position]);
}
?>
```

程序运行后会显示主界面，主界面含有 4 个列表项，单击列表项可以执行读取短消息总数量、读取短消息所有属性、读取所有短消息内容和删除短消息 4 个操作，运行结果如图 19-2-2 所示，其中图 19-2-2(a)是短消息主界面，图 19-2-2(b)是读取出的所有短消息内容。



(a) 主界面



(b) 读取短消息内容

图 19-2-2 短消息操作

下面是查看短消息的指定属性值的范例，例中在数组 smsAttrs 中设置了短消息属性集，调用函数 smsGetMessageById 返回短消息的指定属性值。

【例 19-2-3】 （代码位置：\19\MsgAppointedAttribute 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="match_parent"
  android:layout_height="match_parent" android:background="#ff000000">
    <TextView
      android:text=" "
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textSize="25px"
      android:id="@+id/TextView01"
    />
    <ListView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/ListView01"
      android:drawSelectorOnTop="false"
    />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
    require_once("Android.php");
    $droid=new Android();
    $ret=$droid->smsGetMessageIds(false);
    $ret=$ret['result'];
    $sms=array();
    foreach($ret as $key=>$value)
    {
        $tmp=$droid->smsGetMessageById($value);
        $body=$tmp['result']->body;
        $address=$tmp['result']->address;
        $sms[]="短消息内容：$body \n 发信号码：$address";
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetProperty("TextView01","text","查看短消息属性");
    $droid->fullSetList("ListView01",$sms);
    //清空消息队列和等待用户点击列表事件
    $droid->eventClearBuffer();
    $event=$droid->eventWait();
    while($event['result']->name !='itemclick')
          $event=$droid->eventWait();
    //获取用户点击的列表项
    $item=$event['result']->data;
    $item=$item->position;
    $ListIds=$droid->smsGetMessageIds(false);
    $ListIds=$ListIds['result'];          //所有短消息的 id 列表
    $id=$ListIds[$item];                  //获取用户点击列表项所对应短消息的 id
    //查看短消息的指定属性
    $smsAttrs=array();
    $smsAttrs[]="_id";
    $smsAttrs[]="thread_id";
    $smsAttrs[]="address";
    $smsAttrs[]="person";
    $smsAttrs[]="date";
    $smsAttrs[]="protocol";
    $smsAttrs[]="read";
    $smsAttrs[]="status";
    $smsAttrs[]="type";
    $smsAttrs[]="subject";
    $smsAttrs[]="body";
    $smsAttrs[]="service_center";
```

```
    $smsAttrs[]="locked";
    $smsAttrs[]="error_code";
    $smsAttrs[]="seen";
    $smsAttrs=$droid->smsGetMessageById($id,$smsAttrs);
    $smsAttrs=$smsAttrs['result'];
    $smsAttrsList=array();
    foreach($smsAttrs as $key=>$value)
    {
        $smsAttrsList[]="$key: $value";
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetProperty("TextView01","text","短消息属性");
    $droid->fullSetList("ListView01",$smsAttrsList);
    $droid->eventClearBuffer();
    $droid->eventWait();
?>
```

程序运行后以列表方式显示所有短消息，用户可以单击列表项查看短消息的所有属性，运行结果如图 19-2-3 所示，图 19-2-3(a)是短消息列表，图 19-2-3(b)是短消息的属性和属性值。



(a) 短消息列表



(b) 属性列表

图 19-2-3　查看短消息属性

19.3　WiFi 和蓝牙

19.3.1　WiFi 简述

带 WiFi 无线上网功能的笔记本电脑在工作、生活、学习中的应用已经相当普遍，它的方便性也得到了众人的认可。而随着娱乐、商务便携化需求的提升，越来越多的手机开始内置 WiFi 功能，这样手机也可以像笔记本电脑那样轻松接入无线网络了。在要求不是太高的情况下，用手机 WiFi 上网实现收发邮件、上网查资料、QQ/MSN 通信交流和看在线电影等应用都不是难事。

WiFi 就是一种无线联网技术，常见的是使用无线路由器，在这个无线路由器的信号覆盖的范围内都可以采用 WiFi 连接的方式进行联网。如果这个无线路由器连接了一个 ADSL 线路或其他的联网线路，则又称为连接热点。没有连接热点，或者有连接热点但是没有这个连接热点的密码，WiFi 终端设备是无法使用这个连接热点上网的。

Android 对 WiFi 有较好的支持，应用可以通过 WiFi 上网从而节省流量。但由于 WiFi 非常耗量，为了省电，Android 的 WiFi 加了一个休眠策略，可以设置永远不断开、充电时不断开和锁屏时断开。默认情况下当屏幕被关掉以后，如果没有应用程序在使用 WiFi，WiFi 会在 2min 后进入睡眠状态。如果应用程序想在屏幕被关掉后继续使用 WiFi 则需要锁住 WiFi，该操作会阻止 WiFi 进入睡眠状态，当应用程序不再使用 WiFi 时应该释放 WiFi，让 WiFi 可以进入睡眠状态以节省电源。要在应用中使用 WiFi 功能，开发人员需要掌握 WiFi API。

19.3.2 WiFi API

```
checkWifiState()
```

checkWifiState 函数检测 WiFi 状态，如果 WiFi 可用则返回 true。

```
toggleWifiState(Boolean enabled[optional])
```

toggleWifiState 函数打开或者关闭 WiFi，如果 WiFi 可用则返回 true。

```
wifiDisconnect()
```

wifiDisconnect 函数从当前连接的 WiFi 网络断开，操作成功则返回 true。

```
wifiGetConnectionInfo()
```

wifiGetConnectionInfo 函数返回当前连接热点的信息，包括接入点的 IP 地址、接入点的名称、用户身份、连接速度、网络 ID 和客户端状态等信息。

```
wifiGetScanResults()
```

wifiGetScanResults 函数返回当前可连接的 WiFi 连接列表。

```
wifiLockAcquireFull()
```

wifiLockAcquireFull 函数获取一个完整的 WiFi 锁。

```
wifiLockAcquireScanOnly()
```

wifiLockAcquireScanOnly 函数获取一个只搜索 WiFi 的锁。

```
wifiLockRelease()
```

wifiLockRelease 函数释放一个之前获得的 WiFi 锁。

```
wifiReassociate()
```

wifiReassociate 函数重新与当前的接入点关联，操作成功则返回 true。

```
wifiReconnect()
```

wifiReconnect 函数重新连接到当前接入点，操作成功则返回 true。

```
wifiStartScan()
```

wifiStartScan 函数开始搜索 WiFi 接入点，如果扫描初始化成功则返回 true。

19.3.3 WiFi 范例

下面是 WiFi 的范例，范例先使用函数 wifiGetConnectionInfo 获取连接热点信息，然后以列表形式显示信息。

【例 19-3-1】 （代码位置：\19\Wifi 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="match_parent"
  android:layout_height="match_parent" android:background="#ff000000">
    <TextView
        android:text="连接热点信息"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25px"
        android:id="@+id/TextView01"
    />
    <ListView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/ListView01"
      android:drawSelectorOnTop="false"
    />
</LinearLayout>
```

文件 test.php 内容如下：

```
<?php
    require_once("Android.php");
    $droid=new Android();
    $attrs=$droid->wifiGetConnectionInfo();
    $attrs=$attrs['result'];
    $AttrsList=array();
    foreach($attrs as $key=>$value)
        $AttrsList[]="$key: $value";
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
```

```
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$AttrsList);
    $droid->eventClearBuffer();
    $droid->eventWait();
?>
```

程序需要在真实手机上运行,而且手机还要支持 WiFi。尽管模拟器并不支持 WiFi,但该程序在其上运行并不会出错,只是连接热点信息内容无效。

19.3.4 蓝牙简述

所谓"蓝牙"(bluetooth)技术,是一种支持设备短距离通信(一般是 10~30m)的无线电技术。利用蓝牙技术,能够有效地简化掌上电脑、笔记本电脑、移动电话和手机等移动通信终端设备之间的通信,也能够成功地简化以上这些设备与因特网之间的通信。蓝牙产品中有蓝牙适配器(bluetooth dongle)、蓝牙耳机(bluetooth headset)、蓝牙车载设备(bluetooth car kit)、蓝牙 GPS、蓝牙鼠标键盘(bluetooth mouse&key)和蓝牙立体声 MP3 等。

Android 从 SDK 2.0 开始支持蓝牙,但蓝牙应用须在设备上运行,不支持模拟器。要进行蓝牙通信需要完成下面 4 个步骤:启动蓝牙;发现已经配对或者附近可用的蓝牙设备;连接设备;在不同设备之间传输数据。

19.3.5 蓝牙 API

```
bluetoothAccept(String uuid,Integer timeout)
```

bluetoothAccept 函数监听并接受一个蓝牙连接,阻塞直到连接建立或者连接失败。参数 uuid 是通用唯一识别码,该值可选,默认值是 457807c0-4897-11df-9879-0800200c9a66;参数 timeout 表示监听超时时间,值 0 表示永远监听,该参数可选,默认值是 0。

```
bluetoothActiveConnections()
```

bluetoothActiveConnections 函数返回一个活动的蓝牙连接。

```
bluetoothConnect(string uuid,string address)
```

bluetoothConnect 函数通过蓝牙与一个移动终端进行连接,阻塞直到连接建立或者连接失败,连接建立成功则返回 true。参数 uuid 见前面介绍,此处的 uuid 必须与服务端使用的 uuid 相匹配;参数 address 是可选的,如果未提供此参数,将向用户显示一个已发现的可连接的设备列表供用户选择。

```
bluetoothDiscoveryCancel()
```

bluetoothDiscoveryCancel 函数取消搜索蓝牙设备进程,也就是说当正在搜索设备时调用这个方法将不再继续搜索,成功取消则返回 true,否则返回 false。

```
bluetoothDiscoveryStart()
```

bluetoothDiscoveryStart 函数开始搜索蓝牙设备发现进程，成功则返回 true，否则返回 false。

```
bluetoothGetConnectedDeviceName(String connID)
```

bluetoothGetConnectedDeviceName 函数获取已连接的设备名称。参数 connID 表示连接 id，该参数可选，默认值是 null。

```
bluetoothGetLocalAddress()
```

bluetoothGetLocalAddress 函数获取本地蓝牙设备的硬件地址。

```
bluetoothGetLocalName()
```

bluetoothGetLocalName 函数获取蓝牙可见设备的名称。

```
bluetoothGetRemoteDeviceName(String address)
```

bluetoothGetRemoteDeviceName 函数获取远程蓝牙设备的名称。参数 address 表示远程蓝牙设备地址。

```
bluetoothGetScanMode()
```

bluetoothGetScanMode 函数获取蓝牙搜索模式，返回值为－1 表示蓝牙不可用，0 表示蓝牙不可发现且不可连接，1 表示可连接但不可发现，3 表示可连接可发现。

```
bluetoothMakeDiscoverable(integer duration)
```

bluetoothMakeDiscoverable 函数设置蓝牙连接在一段时间内为可见状态。参数 duration 是移动终端蓝牙可见时间，是可选的，默认值是 300，以秒为单位。

```
bluetoothRead(integer buffersize,String connID)
```

bluetoothRead 函数读取指定长度的数据。长度由 buffersize 参数指定，该参数可选，默认值为 4096；参数 connID 是连接 id，函数返回 ASCII 编码的字符串。

```
bluetoothReadBinary(Integer bufferSize,String connID)
```

bluetoothReadBinary 函数读取指定长度的数据。长度由 buffersize 参数指定，该参数可选，默认值为 4096；参数 connID 是连接 id，函数返回 Base64 编码的字符串。

```
bluetoothReadLine(String connID)
```

bluetoothReadLine 函数读取下一行数据。参数 connID 表示连接 id，该参数可选，默认值为 null。

```
bluetoothReadReady(String connID[optional,default]:Connection id)
```

bluetoothReadReady 函数判断是否全部读取完成，如还有后续数据可供读取则返回 true。

```
bluetoothStop(String connID)
```

bluetoothStop 函数停止蓝牙连接。参数 connID 是连接 id,可选,默认值为 null。

```
bluetoothWrite(String ascii,String connID)
```

bluetoothWrite 函数通过当前打开的蓝牙连接传送 ASCII 编码数据。参数 bytes 表示要传送的数据;参数 connID 是连接 id,可选,默认值是 null。

```
checkBluetoothState()
```

checkBluetoothState 函数检测蓝牙连接的状态,如果蓝牙可用则返回 true。

```
toggleBluetoothState(Boolean enabled,Boolean prompt)
```

toggleBluetoothState 函数打开或者关闭蓝牙,如果蓝牙可用,则返回 true。参数 enabled 是布尔值,表示是否打开或关闭蓝牙,值 true 表示打开,值 false 表示关闭,该参数可选,默认值是 true;参数 prompt 是布尔值,表示是否提示用户当前蓝牙状态已改变,值 true 表示提示,值 false 表示不提示,该参数可选,默认值是 true。

19.4 MediaPlayerFacade

19.4.1 Android 多媒体简述

移动设备越来越多地用作多媒体设备,很多 Android 设备都内置了相机、麦克风和扬声器,能让用户播放和录制各种格式的多媒体。在应用程序中使用多媒体可以显著地增强应用程序的吸引力、实用性和功能性。Android SDK 提供了全面的多媒体支持,让开发人员能够在应用程序中集成音频和视觉媒体(静态图像和视频)。如果要录制音、视频或拍摄视频,要在真实手机上进行测试运行,而且要确保设备硬件支持摄像头和麦克风等。

在音频方面,Android 系统默认支持 MP3 和 AAC 两种音频编码。MP3(Moving Picture Experts Group Audio Layer Ⅲ)是一种音频压缩技术,它被设计用来大幅度地降低音频数据量。利用 MP3 技术,将音乐以 1∶10 甚至 1∶12 的压缩率压缩成容量较小的文件,而对于大多数用户来说,重放的音质与最初的不压缩音频相比没有明显的下降。AAC(Advanced Audio Coding)是一种专为声音数据设计的文件压缩格式,它是在 MP3 的基础上发展而来的,相对于 MP3,AAC 具有音质更佳和文件更小的优点。

在视频方面,Android 系统默认支持 MP4 和 H.264 视频编码。

如果要想让系统支持更多格式的媒体,开发人员需要自行扩展编码功能,本书不深入探讨这个问题,有兴趣的朋友可以自行参看其他资料。如果要让应用支持多媒体功能,需要借助 MediaPlayer API 来实现,它提供了播放、暂停、停止、定位和查看媒体信息等功能。

19.4.2 MediaPlayer API

```
mediaIsPlaying(String tag)
```

mediaIsPlaying 函数检查媒体文件是否在播放，如果正在播放则返回 true。参数 tag 表示媒体标识符，该参数可由用户来指定，该参数可选，默认值是 default。

```
mediaPlay(String url,String tag,Boolean play)
```

mediaPlay 函数打开一个媒体文件，如果媒体文件能打开而且可以播放则返回 true。参数 url 表示媒体资源的 URL；参数 tag 表示媒体标识符，可选，默认值是 default；参数 play 是布尔值，表示是否立即播放，可选，默认值是 true。

```
mediaPlayClose(String tag)
```

mediaPlayClose 函数关闭指定的媒体文件，关闭成功则返回 true。参数 tag 表示媒体标识符，可选，默认值是 default。

```
mediaPlayInfo(String tag)
```

mediaPlayInfo 函数查询当前媒体的信息，查询成功则返回媒体信息，媒体信息由固定的属性组成，具体的属性和含义如表 19-4-1 所示。

表 19-4-1　媒体信息属性

属　　性	含　　义
tag	媒体播放器标识符
loaded	标识媒体是否已加载，值 true 表示已加载，否则表示没有元素被加载
duration	媒体的播放长度，单位为毫秒
position	当前播放位置，单位为毫秒，位置可由函数 mediaPlaySeek 控制
isplaying	表示媒体是否正在播放，值 true 表示正在播放，否则未播放，媒体是否播放可由函数 mediaPlayPause 和 mediaPlayStart 控制
url	打开媒体文件的 URL
looping	媒体是否循环播放，值 true 表示循环，可由函数 mediaPlaySetLooping 控制

```
mediaPlayList()
```

mediaPlayList 函数获取当前已加载媒体的列表，返回媒体列表，列表项包含有媒体的 Tag 标认符。

```
mediaPlayPause(String tag)
```

mediaPlayPause 函数暂停媒体文件的播放，暂停成功则返回 true。参数 tag 表示媒体标识符，可选，默认值是 default。

```
mediaPlaySeek(Integer msec,String tag)
```

mediaPlaySeek 函数定位媒体播放位置，返回值是新的定位位置。参数 msec 是新位置，单位为毫秒，该值应该在 0 到媒体最大位置之间；参数 tag 表示媒体标识符，可选，默认值是 default。

```
mediaPlaySetLooping(Boolean enabled,String tag)
```

mediaPlaySetLooping 函数设置是否循环播放媒体，如果成功设置则返回 true。参数 enabled 表示是否循环播放，值 true 表示循环，参数可选，默认值是 true；参数 tag 表示媒体标识符，可选，默认值是 default。

```
mediaPlayStart(String tag[optional,default default]: string identifying resource)
```

mediaPlayStart 函数立即开始播放媒体文件，成功则返回 true。参数 tag 表示媒体标识符，可选，默认值是 default。

19.4.3 简易多媒体播放器范例

下面是简易多媒体播放器的范例，播放器会自动加载/sdcard/sl4a/scripts/media/目录中的多媒体文件，用户可以选择相应的按钮播放媒体、停止媒体、查看媒体信息和查看播放列表。

【例 19-4-1】 （代码位置：\19\Media 目录）

文件 mylayout.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent" android:layout_height="fill_parent"
  android:background="#ff000000" >
      <TextView
          android:text="简易媒体播放器"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:textSize="25px"
          android:id="@+id/TextView01"
        />
      <TextView
          android:text="当前媒体："
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:id="@+id/TextView02"
        />
    <SeekBar
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="10dp"
        android:max="100"
        android:progress="0"
        android:id="@+id/Progress01"
```

```
    />
  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="horizontal"
  android:layout_width="fill_parent"
  android:background="#ff000000" android:gravity="center">
      <Button
        android:text="播放下一首"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/play"
        />
      <Button
        android:text="停止"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/stop"
        />
      <Button
        android:text="媒体信息"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/viewCurrentMedia"
        />
      <Button
        android:text="媒体列表"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/viewList"
        />
      <Button
        android:text="退出"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/exit"
        />
  </LinearLayout>
</LinearLayout>
```

文件test.php内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
$filenames=array();
$dir=dir("/sdcard/sl4a/scripts/media");
```

```
//加载/sdcard/sl4a/scripts/media目录中的文件
while(($file=$dir->read())!==false)
{
    if(($file!='.')&&($file!='..'))
        $filenames[]="/sdcard/sl4a/scripts/media/" . $file;
}
$dir->close();
$layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
$droid->fullShow($layout);
$flag=true;
while($flag)
{
    $event=$droid->eventWait(1000);
    $event=$event['result'];
    if($event->name=='click')
    {
        switch($event->data->id)
        {
            case "play":
                stopMedia($droid);
                //播放下一首媒体
                $music=next($filenames);
                if(  $music==false)
                    $music=reset($filenames);

                $droid->mediaPlay($music,"default");
                //初始化播放进度条和更新当前媒体文件
                $droid->fullSetProperty("Progress01","progress","0");
                $droid->fullSetProperty("TextView02","text","当前媒体: $music");
                break;
            case "stop":
                stopMedia($droid);
                break;
            case "viewCurrentMedia":
                viewCurrentMedia($droid);
                break;
            case "viewList":
                viewList($droid,$filenames);
                break;
            case "exit":
                stopMedia($droid);
                $flag=false;
                break;
            default:
```

```
                break;
            }
        }
        if($droid->mediaIsPlaying())
        {   //更新播放进度条
            $info=$droid->mediaPlayInfo("default");
            $info=$info['result'];
            $progress=0;
            $position=$info->position;
            $duration=$info->duration;
            $progress=(int)($position/$duration*100);
            echo $progress;
            $droid->fullSetProperty("Progress01","progress","$progress");
        }
}
function stopMedia($droid)
{
        //功能:关闭当前已加载的媒体
        $info=$droid->mediaPlayInfo("default");
        $$info=$info['result'];
        if(isset($info->loaded)==false)
            $droid->mediaPlayClose("default");
}
function viewCurrentMedia($droid)
{   //功能:查看当前媒体信息
    $info=$droid->mediaPlayInfo("default");
    $info=$info['result'];
    $droid->dialogCreateAlert("媒体属性");
    $attrsList=array();
    foreach($info as $key=>$value)
      $attrsList[]="$key: $value";
    $droid->dialogSetItems($attrsList);
    $droid->dialogShow();


}
function viewList($droid,$filenames)
{   //功能:查看媒体列表
    $droid->dialogCreateAlert("媒体播放列表");
    $musicList=array();
    foreach($filenames as $key=>$value)
    $musicList[]="序号$key: $value";
    $droid->dialogSetItems($musicList);
    $droid->dialogShow();
}
```

?>

先把多媒体文件复制到模拟器或手机上的/sdcard/sl4a/scripts/media/目录下,确保多媒体文件是 MP3 或 MP4 格式,如果格式不正确将不能播放媒体文件。然后再运行程序,用户单击"播放"按钮将播放下一个媒体文件,"当前媒体"处将更新显示当前正在播放的媒体文件,同时,播放进度条会随时间的变化而前进,如果是最后一个媒体文件则播放第一个媒体文件;单击"停止"按钮将停止播放当前媒体文件;单击"媒体信息"按钮将以列表方式显示当前媒体的信息属性和属性值;单击"媒体列表"按钮将以列表方式显示所有要播放的媒体文件;单击"退出"按钮将结束程序。程序运行结果如图 19-4-1 所示。



图 19-4-1 媒体播放器

19.5 语音合成和语音识别 API

19.5.1 语音合成简述

Text To Speech(TTS),翻译为"从文本到语音",即语音合成,它是中文信息处理领域的一项前沿技术,解决的主要问题就是如何将文字信息转化为可听的声音信息,而通过计算机语音合成则可以在任何时候将任意文本转换成具有高自然度的语音。

Android 从 1.6 版本开始支持 TTS,目前市场上支持 Android 的 TTS 产品很多,如 Pico、科大讯飞、捷通华声、InfoTalk、微软和开源项目 eyes-free 的 TTS 产品。TTS 产品可分为两大类:第一类是与系统接口适配;第二类是与系统接口不适配。第一类的做法是把 TTS 引擎与语言包数据分开,Pico、科大讯飞和 eyes-free 等属于这一类。这类做法的好处是可以通过系统提供的接口使用 TTS 功能,易实现多国语言拓展,但不足的是,设置语音朗读角色和设置语速快慢就得通过系统的接口。第二类把引擎和语言包打包成一个 APK 包,在安装完成之后,应用不是通过系统提供的接口而是通过指定的接口实现语音合成功能,捷通华声等属第二类做法。

Android 内置默认使用的是 Pico 语音合成引擎,但是只支持英语、法语、德语、意大利语和西班牙语 5 种语言,暂时没有对中文提供支持。因此使用 Android 默认的 TTS 引擎是没法朗读中文的。

开源项目 eyes-free(http://code.google.com/p/eyes-free/)除了提供 Pico 外,还把支持其他更多语言语音合成的另一个 TTS 引擎 eSpeak 也移植到了 Android 平台,其中就支持中文的语音合成。因此在安装了 eyes-free 提供的 TTS Service Extende 的 APK 后,就可以在程序中使用 eyes-free 提供的 TTS library,并把 TTS 引擎设置为不是默认的 Pico,而是 eSpeak,就可以实现朗读中文了。不过经过测试,朗读中文的实际效果还是很差的,只能说勉强可以朗读而已。

讯飞语音 TTS 1.0 是讯飞科技公司新推出的一款 TTS 语音合成 Android 软件,其

中文语音包包括一个中文女声和一个中文男生，发音为标准的大陆普通话，效果远超三星 TTS 和 SVOX。安装设置好讯飞 TTS 后，就可以自动应用于需要中文发音的软件了，如静读天下和短信播报等。如果安装了讯飞语音 TTS 而不能顺利开启语音的话，检查有没有下载这个安装讯飞语音 TTS 中文语音包，讯飞语音 TTS 中文语音包在离线状态下便可以使用。讯飞语音 TTS 主程序和语音包可从网站 http://ifly-tts.softonic.cn/android/下载，其安装如下：

（1）安装讯飞 TTS 主程序 APK 和讯飞语音包 APK。

（2）选择"系统设置"→"语音输入输出设置"→"文字转语音设置"→"讯飞语音合成"菜单命令，系统的默认引擎将使用讯飞语音合成，引擎使用的语言和语速也可以通过其中的选项进行设置。

19.5.2 语音合成 API

Android 系统提供了两个语音合成函数，一个用于检测系统的语音合成引擎是否在发声，另一个调用语音合成引擎发声。

```
ttsSpeak(String message)
```

ttsSpeak 函数通过 TTS 读出参数 message 包含的文本内容。

```
ttsIsSpeaking()
```

ttsIsSpeaking 函数检测系统是否在使用 TTS，在使用则返回 true。

下面是语音合成 API 的范例。

【例 19-5-1】 （代码位置：\19\tstTTS.php）

```
<?php
require_once("Android.php");
$droid=new Android();
$droid->ttsSpeak("你好");
$event=$droid->eventWait(2000);
$event=$event['result'];
?>
```

程序运行后将念出"你好"。如果使用的是系统默认的 Pico 引擎则不能正确发音，如果使用的是讯飞引擎则能以普通话发出正确的声音。注意，使用讯飞引擎必须要安装讯飞中文语音合成库并设置为系统默认使用的引擎，否则不能发音或不能正确发音。

19.5.3 短消息朗读范例

手机原先是为视力正常的人设计的，没有充分考虑到一些视力较弱的人或盲人的需要。如果在来电显示、阅读短消息和闹钟提示等功能中加入语音合成，则可以帮助弱视人士解决大部分的手机应用困难，相信这样也会吸引广大普通用户的眼球。下面是短消息朗读的范例，范例会以列表形式显示所有短消息，用户点短消息后系统会把短消息内

容朗读出来。

【例 19-5-2】（代码位置：\19\TTSMsg 目录）

文件 mylayout. xml 内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
  android:background="#ff000000"
  >
  <ListView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ListView01"
            android:drawSelectorOnTop="false"
          />
</LinearLayout>
```

文件 test. php 内容如下：

```
<?php
require_once("Android.php");
$droid=new Android();
speakMessage($droid);

function speakMessage($droid)
{ //功能：普通话读短消息
//先置条件：安装和设置好讯飞语音等中文语音合成引擎和语音库
    $ret=$droid->smsGetMessageIds(false);
    $ret=$ret['result'];
    $sms=array();
    foreach($ret as $key=>$value)
    {
        $tmp=$droid->smsGetMessageById($value);
        $body=$tmp['result']->body;
        $address=$tmp['result']->address;
        $sms[]="短消息内容：$body \n 发信号码：$address";
    }
    $layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
    $droid->fullShow($layout);
    $droid->fullSetList("ListView01",$sms);
    //等待点击列表项
    $droid->eventClearBuffer();
    $event=$droid->eventWait();
```

```
        //获取点击的短消息并读出短消息
        $item=$event['result']->data;
        $item=$item->position;
        $droid->ttsSpeak($sms[$item]);
        $droid->ttsIsSpeaking();
        //检测是否在读文本
        do
        {
            $droid->eventWait(1000);
            $flag=$droid->ttsIsSpeaking();
            $flag=$flag['result'];
        } while($flag==true);
    }
    ?>
```

程序运行结果如图19-5-1所示，用户单击短消息后系统将朗读该条短消息，直到朗读结束，程序才结束并退出。



图 19-5-1　朗读短消息

参 考 文 献

[1] Ed Burentte. Android 基础教程(第 3 版 · 修订版)[M]. 田俊静,张波,黄湘情,等译. 北京：人民邮电出版社,2011.

[2] 李卉,张魏,祝延彬. Android 移动开发一本就够[M]. 北京：人民邮电出版社,2012.

[3] 袁国忠. Android 应用开发入门经典[M]. 2 版. 北京：人民邮电出版社,2012.

[4] 梁文新,宋强,刘凌霞. Ajax+PHP 程序设计实战详解[M]. 北京：清华大学出版社,2010.

[5] rjmatthews62. API Reference (release R6) [EB/OL]. http://code. google. com/p/android-scripting/wiki/ApiReference , 2013-12-23.

[6] 熊猫 82. 浅谈 Android 五大布局(一)——LinearLayout、FrameLayout 和 AbsoulteLayout [EB/OL]. http://www. cnblogs. com/wisekingokok/archive/2011/08/23/2150452. html,2013-1-2.

[7] 熊猫 82. 浅谈 Android 五大布局(二)——RelativeLayout 和 TableLayout[EB/OL]. http://www. cnblogs. com/wisekingokok/archive/2011/08/24/2152004. html, 2013-1-2.

[8] 郭靖. ASP. NET 开发技术大全[M]. 北京：清华大学出版社,2009.

[9] 杨宇. PHP 典型模块与项目实战大全[M]. 北京：清华大学出版社,2011.

[10] lipub...@ gmail. com. 中文版 api[EB/OL]. http://code. google. com/p/sl4a-chinese/wiki/ApiReference , 2013-11-11.

[11] 廖雪峰. JSON 入门指南[EB/OL]. http://www. ibm. com/developerworks/cn/web/wa-lo-json/, 2013-10-12.

[12] Steve. XML 作用[EB/OL]. http://www. blogjava. net/xxxzheng/archive/2005/12/02/22270. html, 2013-6-8.

[13] 磊子. PHP 技术分享[EB/OL]. http://www. favortt. com/category/php-technology-share, 2013-8-10.

[14] 曹建. WAP 编程与开发实例教程[M]. 北京：电子工业出版社,2001.

[15] 汪翔,张静. WAP 建站技术详解与实例[M]. 北京：清华大学出版社,2001.

[16] 杜江. PHP 5 与 MySQL 5 Web 开发技术详解[M]. 北京：电子工业出版社,2007.